smart engines™

Whitepaper

Binarization Algorithms
for Documents Recognition

Binarization algorithms for Documents Recognition

During the 9-year history of International Competition on Document Binarization DIBCO17 held within ICDAR conference, a lot of bold and unconventional algorithms of binarization have been proposed. Although our team at the Smart Engines does not normally use such algorithms in mobile recognition, we were convinced that our product had unique features worth presenting to the public, and we decided to participate for the first time.

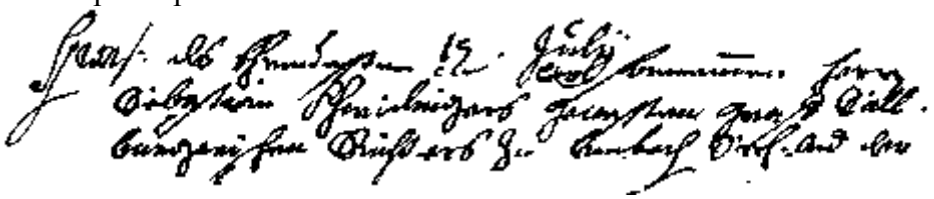


Figure 1: Document image binarization results assembled into gif animation

In essence, contestants were presented with a plethora of original document images (see Figure 2) with corresponding to their ideal binary results (according to human-made annotation) (see Figure 3). The task was to come up with an algorithm that would transform the initial images to two tier black-and-white images as precisely as possible. Several metrics were evaluated and considered to judge each team's performance. It is worth mentioning, that teams had no access to the "ideal" images beforehand. Thus, they could only calibrate their algorithms using the previous years' images. The test dataset consisted of complex images with fine watercolors, see-through symbols, etc. - so that the competition could distinguish truly efficient algorithms.

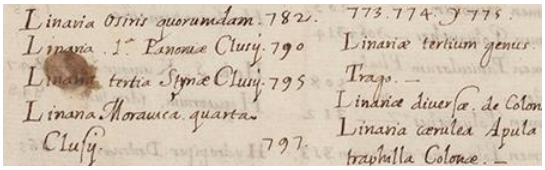


Figure 2: Example of an original document image on a complex background

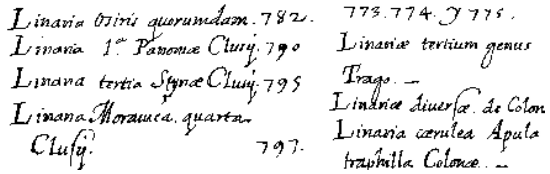


Figure 3: Example of an expected ideal document image binary result

Solution Framework

First, all the data from the previous competitions has been collected: 65 written and 21 printed pictures in total. Obviously, in order to achieve the best results, not only did we inspect available images from the competition, but also looked for open data sets and thousands archived printed and handwritten images. Having inspected the images, it became clear to us what kind of challenges we could potentially face and which had been neglected by the competition organizers. For instance, documents from the past competitions had never contained grid elements, despite occurrence of tables in archives.

During the preparation for the competition, we tried to follow several parallel ways. Not only did we use well-understood classical algorithmic approaches, but also employed machine-learning methods for "object-background" pixel classification, despite the lack of initially presented to us sets of data. Since this approach turned out to be the most efficient, let us explore it in more details.

Neural Network Architecture

Initially U-net has been chosen as the neural network architecture type, since such an architecture has proven itself during numerous competitions related to segmentation (dstl, Ultrasound Nerve Segmentation, Data Science Ball 2017). Furthermore, a large class of well-known binarization algorithms are particularly coherent in such an architecture or similar ones.

An important advantage of this architecture is that, in pursuit of training the network, one can create sufficient training material out of small amount of available images. Besides, the network has a

relatively small number of weights. There are certain nuances, though. The artificial neural network that is used does not, strictly speaking, resolve the problem of binarization. Essentially, each pixel is matched to some number in the range between 0 and 1, which characterises to what extent the pixel belongs to one of the classes (meaningful filing or background) and which needs to be transformed into the final result.
80% of source images were taken as a training sample, whereas the rest (20%) of them were devoted to validation and testing. Colored images were transformed to greyscale to avoid overfitting, then they were divided into non-overlapping pixel windows of size 128x128 (see Figure 4). The dimensions of the windows had been chosen empirically, by testing windows sized from 16x16 up to 512x512 pixels. We obtained 70000 windows out of 100 initial images, the former later were used as a neural network input.

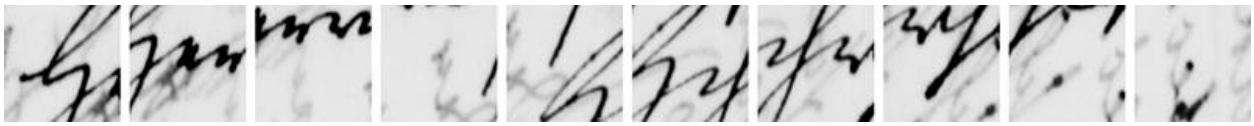


Figure 4: Example of a greyscale document image divided into pixel windows

Each window was then matched to a binary mask. We followed the principle of carefully examining and understanding the underlying process, as opposed to merely launching hyperopt for a week. Adam was chosen for stochastic optimization and cross-entropy as a loss function metric.

Initial experiments

The first experiments showed vividly that this approach enables to achieve superior results compared with simple non-trainable methods (like Otsu or Niblack). The neural network was easily trainable, and the process shortly converged to the acceptable optimum.
Each animation was obtained as follows: during the training process, as the quality improved, the network would receive the same image over and over and the obtained results were assembled into one gif animation (see Figure 6).

Figure 5: The original handwritten document image on a complex background

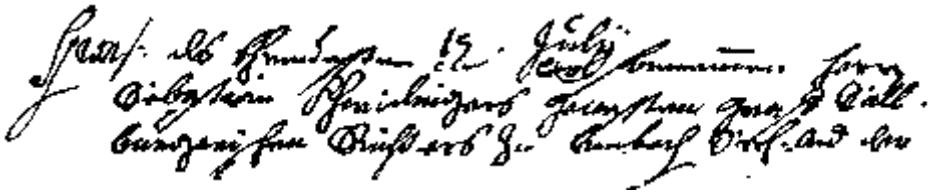


Figure 6: Obtained binarization results assembled into gif animation

The difficulty of binarization employing this method is due to the fact that occasionally it is hard to distinguish between the background and the specificities of intricate handwriting: blots, certain

parts of the letters are blurred and text from the other side of a paper appear. The one who wrote the manuscript on Figure 5 must have not been the neatest person of his time.

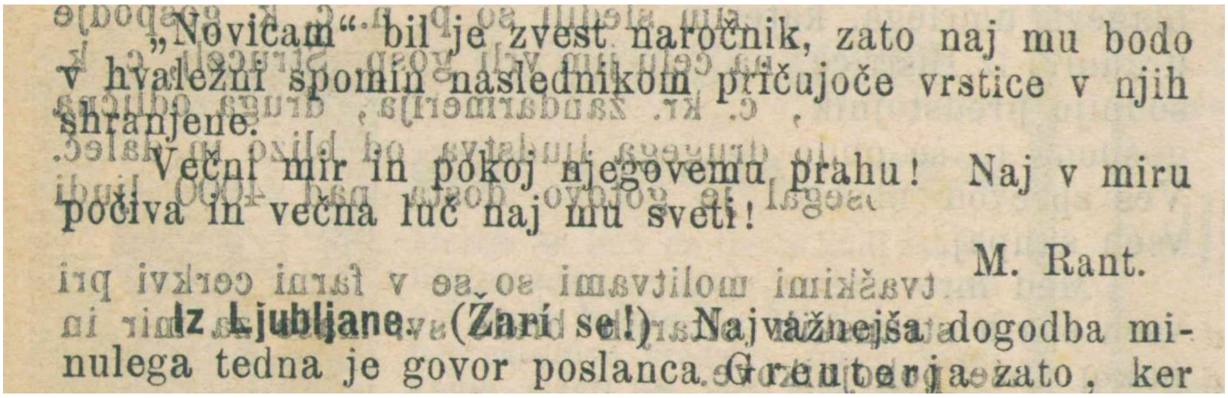
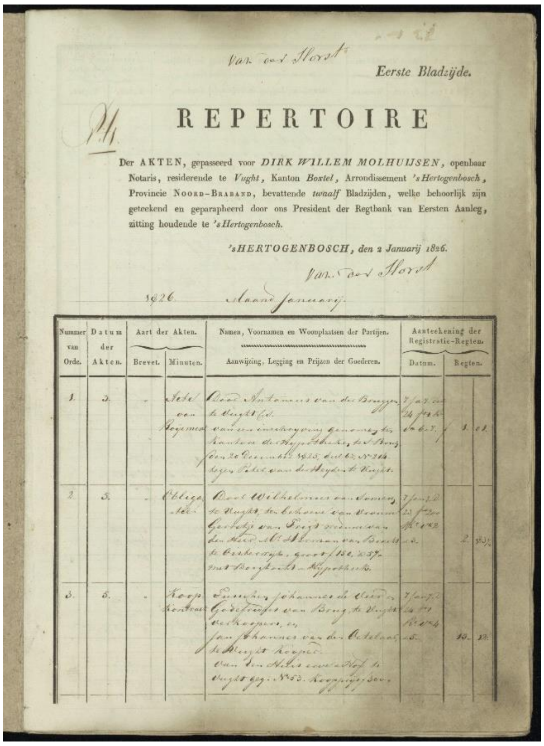

„Novičam“ bil je zvest naročnik, zato naj mu bodo v hvaležni spomin naslednikom pričujoče vrstice v njih shranjene.
Večni mir in pokoj njegovemu prahu! Naj v miru počiva in večna luč naj mu sveti!
M. Rant.
Iz Ljubljane. (Žarí se!) Najvažnejša dogodba minulega tedna je govor poslanca Grenberja zato, ker

Figure 7: Example of a document image with non-uniform background on a see-through paper

In the example shown in Figures 7, apart from the non-uniform background, the text from the other side of the paper can be seen, and the possible way of classifying this text as a background is spotting mirrored characters.

After each experiment, we would also assess the relevancy of each model to the numerous archives from the open database. What we found out, was that applying the network to inputs from the available database would occasionally bring about unsatisfying results. Hence, some of the documents we would add to the training sample. Edges of pages and their markups in particular presented difficulties for us. Five additional documents in total were selected, all of which contained objects of interest.
In the example shown in Figures 8 and 9, besides the misinterpreted edges of the page, both the table and the text in the middle are deficiently defined.


Eerste Bladzijde.
REPERTOIRE
Der AKTEN, gepasseerd voor DIRK WILLEM MOLHUIJSEN, openbaar Notaris, residerende te Vught, Kanton Boxtel, Arrondissement 's Hertogenbosch, Provincie Noord-Braband, bevattende twaalf Bladzijden, welke behoorlijk zijn geteekend en geparapheerd door ons President der Regtbank van Eersten Aanleg, zitting houdende te 's Hertogenbosch.
'sHERTOGENBOSCH, den 2 Januarij 1816.

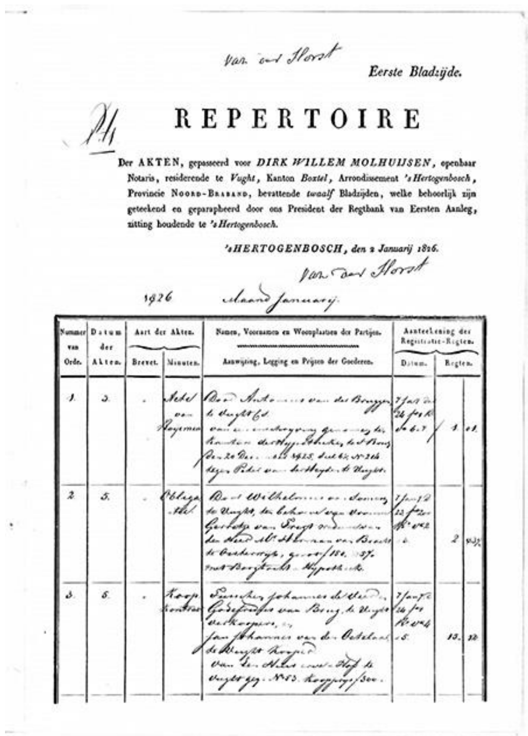
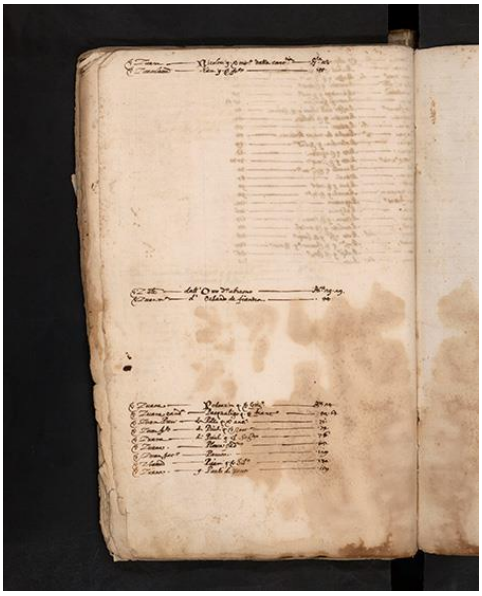

Eerste Bladzijde.
REPERTOIRE
Der AKTEN, gepasseerd voor DIRK WILLEM MOLHUIJSEN, openbaar Notaris, residerende te Vught, Kanton Boxtel, Arrondissement 's Hertogenbosch, Provincie Noord-Braband, bevattende twaalf Bladzijden, welke behoorlijk zijn geteekend en geparapheerd door ons President der Regtbank van Eersten Aanleg, zitting houdende te 's Hertogenbosch.
'sHERTOGENBOSCH, den 2 Januarij 1816.

Figure 8: Example of a document image with a table

Figure 9: Binary result of a document image with a table

When looking at Figure 11 one can notice how the network highlights page edges, which is a sign of inaccuracy from the point of view of this competition.



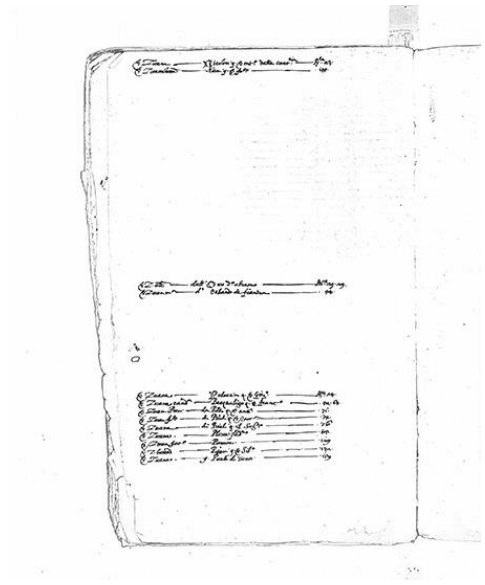


Figure 10: Example of a document image

Figure 11: Document image binarization results where the page edges have been highlighted by the net.

The applied augmentation techniques and how they help

During the process of the network training and the analysis of mistakes for enhancing the quality, several methods of data augmentation have been used, including some distortion methods: reflection of images against the axes, brightness, inversion, noise (Gaussian, salt and pepper), as well as a number of elastic transformations (see example here), variations of image scaling. Each distortion type has been applied due to task specifics, mistakes spotted in the network performance and owing to common practices.

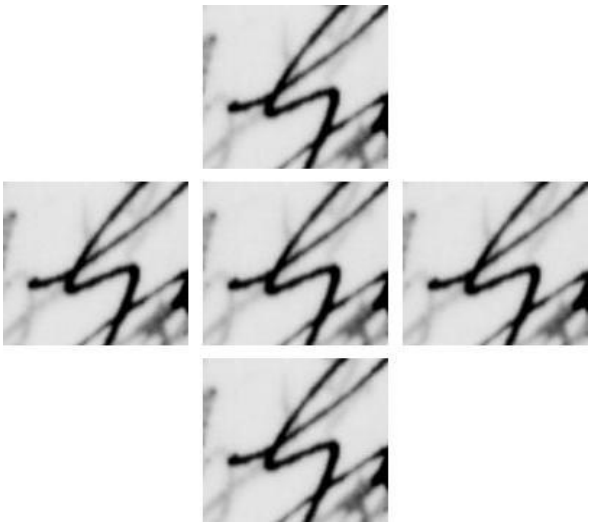


Figure 12: An example of combination of several augmentation methods

In Figure 12 one can see an example of a combination of several augmentation methods, applied on the fly during the training process.

The ensemble process
The next stage of producing the final result is creating an ensemble of several solutions. We have thus used three U-net nets of different architectures, each trained on different data sets and one untrainable binarization method, the latter having been implemented on the edges of images in order to cut them off.

Choosing the final solution
As we progressed to the final version of the algorithm, each step has been subjected to cross-validation in order to make sure we headed in the right direction. The final decision was based on this statistics. It was just one well trained U-net, with the implementation of the following steps: mirrored image, reversed image, image with reduced size, enlarged image.

Results
There were 18 participants from all around the world, including USA, China, India, Middle East countries, and Australia. A lot of solutions were proposed, including the use of neural network models, modification of classic adaptive methods, game theory and combinations of different approaches. Surprisingly enough, although methods used by the participants differed significantly, the final results turned out to be very similar on many occasions. The results for respective methods are presented in the table below

№	Brief Description of the Method	Score	FM	Fps	PSNR	DRD
1	**Our Method (U-net)**	**309**	**91.01**	**92.86**	**18.28**	**3.40**
2	FCN (VGG-like architecture) + post filtering	455	89.67	91.03	17.58	4.35
3	Ensemble of 3-x DSN with a three-level exit, calibrating fragments of different scale	481	89.42	91.52	17.61	3.56
4	Ensemble of 5 FCN — input: fragments of different scale + Howe binary method + RD attributes.	529	86.05	90.25	17.53	4.52
5	Similar to the previous method + CRF post-processing	566	83.76	90.35	17.07	4.33
...	...	...	...	...	...	...
	Otsu		77.73	77.89	13.85	15.5
	Sauvola		77.11	84.1	14.25	8.85

Table 1: DIBCO17 Results Table

Smart Engines team won the competition the first time we participated, and our solutions attained the highest mark on both hand written documents as well as printed ones. Here are a few examples demonstrating the work of our algorithms on a range of text images.