DOI 10.15622/sp.2019.18.6.1381-1406

L. BUREŠ, I. GRUBER, P. NEDUCHAL, M. HLAVÁČ, M. HRÚZ

SEMANTIC TEXT SEGMENTATION FROM SYNTHETIC IMAGES OF FULL-TEXT DOCUMENTS

Bureš L., Gruber I., Neduchal P., Hlaváč M., Hrúz M. **Semantic Text Segmentation from Synthetic Images of Full-Text Documents.**

Abstract. An algorithm (divided into multiple modules) for generating images of full-text documents is presented. These images can be used to train, test, and evaluate models for Optical Character Recognition (OCR). The algorithm is modular, individual parts can be changed and tweaked to generate desired images. A method for obtaining background images of paper from already digitized documents is described. For this, a novel approach based on Variational AutoEncoder (VAE) to train a generative model was used. These backgrounds enable the generation of similar background images as the training ones on the fly.

The module for printing the text uses large text corpora, a font, and suitable positional and brightness character noise to obtain believable results (for natural-looking aged documents).

A few types of layouts of the page are supported. The system generates a detailed, structured annotation of the synthesized image. Tesseract OCR to compare the real-world images to generated images is used. The recognition rate is very similar, indicating the proper appearance of the synthetic images. Moreover, the errors which were made by the OCR system in both cases are very similar. From the generated images, fully-convolutional encoder-decoder neural network architecture for semantic segmentation of individual characters was trained. With this architecture, the recognition accuracy of 99.28% on a test set of synthetic documents is reached.

Keywords: Generation of Synthetic Images, Semantic Text Segmentation, Variational Autoencoder, VAE, Optical Character Recognition, OCR, Aged-Looking Text Generation.

1. Introduction. Computer vision and machine learning fields are fast-growing, innovative, and well-researched fields nowadays. One of the research areas is Optical Character Recognition (OCR), which can be used for reading scanned documents. In this paper, we focus on digitized text documents and its corpus generation – we considered only the documents with typewritten text. The OCR methods are very often used as a follow-up step in the task of textual data digitization. E.g., full-text search, information retrieval, indexing, and document search can be achieved with the data digitalization for digital archiving purposes. Previously, most of OCR systems were rule-based systems, which were usually divided into a few steps – mainly detection and recognition phases. For instance, here are a few; OCRopus [4], GOCR, Ocrad, Tesseract OCR system [21], and many more. All noted systems have been published under the licenses which allow free use. Further, there exists multiple commercial systems, for instance: ABBYY FineReader or OmniPage. These days, methods of machine learning and artificial intelligence are used in the form of Deep Neural Networks (DNN). These approaches are mostly used in the way of end-to-end solutions, which cover all the necessary steps: detection,

localization, and reading [13-15, 24, 25]. The DNNs are successfully used in the area of text detection and recognition in everyday environment domain (e.g., a photo of a street – where text can be located on store signs, car plates, and other areas), so-called wild text reading. In other words, it is a task of detecting text in real-world environments, where text appears in a sparse form (not full-text documents like in our use case). A lot of noise and clutter are present in such data, and the text can be in any font/form/angle/deformation/style. A large amount of training data is required for the training of machine learning methods to be able to generalize the nature of the task. In the case of supervised methods also, the targets need to be available. Generally, one of the most time-consuming tasks in the process of training neural networks is to gather labeled data which are mostly labeled by humans and can be inaccurate. An algorithm for synthetic training data generation, which has been previously published in [10], can be used for the task of wild text reading. In their work, they estimate the depth of the image data. From this information, the geometry is estimated, and the synthetic text was transformed into this detected region to look natural. The algorithms for real-world text reading nowadays achieve good results, but they cannot be used for full-text document reading tasks. It is due to a significant difference in the task itself and in the image/scene appearance. The authors, in the paper [7], published a system for synthetic text generation, and it was for the aim of reading short texts in mobile environments. Our approach is different from theirs in this aspect. Our focus is mainly on full-text document generation, but they generated only short sentences. In this paper, we propose an automatic system for synthetic image data generation – for labeled training dataset creation. The dataset was further used for the training of multiple machine learning algorithms. The data source which we have used are collections of digitized full-text documents from the times after World War II from Czechoslovakia. This data we aim to emulate.

We followed and considerably extended our previously published work; [5] and [9]. The newly obtained results correspond to the ones presented and described by experiments in this paper.

Several algorithms were used in consecutive order to generate aged and natural-looking typewritten documents. In the first step, we automatically extract samples of document's backgrounds – aged-looking paper. We used our real document dataset for generating the empty documents (plain aged-looking paper backgrounds). Next, Variational AutoEncoder (VAE) [16] was trained and fine-tuned from obtained image dataset. Then, the VAE can generate backgrounds from random noise of given properties. Further, for the illusion of aged-looking text, we used brightness and per-character location noise to create even more authentic images. In the final phase, the corresponding text

annotation is automatically generated. Our process is capable of generating a huge amount of text documents – it needs only a text corpus as an input. The process of generation is language independent. Even though we used exact data in our experiments, the proposed system can generate images from any font and background. Only the proper dataset has to be provided. Next, we trained a DNN for semantic segmentation of the characters. During this segmentation process, a label is assigned to every image pixel. The assigning process is explained in [2], where it was used for road segmentation in urban environments. The method was updated and enhanced a few times, you can check [6] – e.g., network architecture, loss function and general approach to decoding were enhanced.

The main contributions of the paper and the described system are:

– an algorithm designed to remove text from the image of the full-text document. This part aims to keep the image as much natural-looking aged-document as possible;

– synthetic background generator based on VAE is trained by image data from the text removal algorithm (aged-looking documents). This output generates artificial/synthetic images – which are then used as background for generating our full-text typewritten documents;

– a system which is capable of generating trustworthy full-text typewritten documents. Moreover, the document layout can be edited during its generation. All layouts have to be defined before the generation starts;

– Full-text images are semantically segmented to provide the first step of an OCR system. Our character recognition accuracy is above 99%.

The paper is organized as follow: in the Introduction describes the topic of this paper and previous related works. As well as in the introduction, we discuss our contributions. Background Extraction section is focused on a text removal algorithm that is capable of obtaining document backgrounds.The way of generating synthetic artificial document backgrounds is described in the Background Generator section.In the next section, the Text Generator is described. With a synthetic background as an input, it is possible to generate an authentic-looking document. Moreover, it is possible to choose a layout of the text.Experiments and an evaluation of the proposed system are described in the next section. The main goal of this section is to verify whether the OCR results have similar accuracy for real and generated data. Semantic Segmentation section is focused on the description of a segmentation algorithm based on fully-convolutional encoder-decoder architecture. Moreover, the process is based on the segmentation of convex hulls. It is a different approach than the direct segmentation of characters.Summary and proposition of future work are discussed in the last section named Conclusion and Future Work.

2. Background Extraction. Our developed algorithm for image background extraction is described in this section. The background extraction algorithm needs digital scans of original typewritten image documents as input. Commonly the text is of a dark shade on a bright background. Our system expects it in this form. This assumption is the truth for the majority of tested free and commercial software (some can handle inverted colors too), but generally, all digitized documents usually meet this requirement. The developed algorithm is based on computing the mean color value, according to this assumption.

The algorithm loads image in RGB and grayscale color format, and the text is found with the use of Otsu's brightness thresholding method [19]. The thresholded binary image is used for localization of corresponding pixels where the text's pixels are stored as ones and non-text's pixels as zeros. The dilatation process with a square kernel of predefined suitable size is applied in order to suppress the effect of color values from the pixels between text and non-text in the mean color computation. The individual mean color of the background's pixels is calculated for every color component separately. The binary image, from the previous step, is used for masking, which removes all the text pixels from the computation. Each of the pixels which belong to a text label is replaced by mean values of the background.

Further steps are applied to our algorithm. The color difference between pixel values and calculated mean color can be large in some image areas – this can leads to brightness discontinuities which are unwanted. The re-computation process enhances the quality of the result by replacing the color in text pixels by a mean value from the text pixel's local neighborhood (NEIGHBOR_MEAN_ADAPT in the Algorithm 1. It creates a desired statistical bias towards the mean background value by using every pixel for the local mean computation, even the already replaced text pixels. The summary of the proposed method is shown in the Algorithm 1. An example of the results of the proposed algorithm is shown in Figure 1.

```
img_color, img_gray ← LOAD_DOCUMENT
binary ← OTSU(img_gray, BINARY_INVERSE)
binary ← DILATE(binary)
red, green, blue ← GET_CHANNELS(img_color)
red_mean ← MEAN(red[binary = 0])
green_mean ← MEAN(green[binary = 0])
blue_mean ← MEAN(blue[binary = 0])
red[binary = 1] ← red_mean
green[binary = 1] ← green_mean
```

```
blue[binary = 1] ← blue_mean
result ← [red, green, blue]
FOR NONZERO_PIXELS AS p IN binary:
   result[p] ← NEIGHBOR_MEAN_ADAPT(result, p, n_size=(k))
SAVE_IMG(result)
```

Algorithm 1. Background extraction algorithm

There are some flaws in the algorithm. When some dark artifacts are present in the input image, the algorithm is not able to remove them completely. Moreover, the procedure occasionally results in high contrast noise in the yielded image. However, these issues are well handled by the VAE method of generating synthetic backgrounds, and thus they do not need to be addressed by this algorithm.

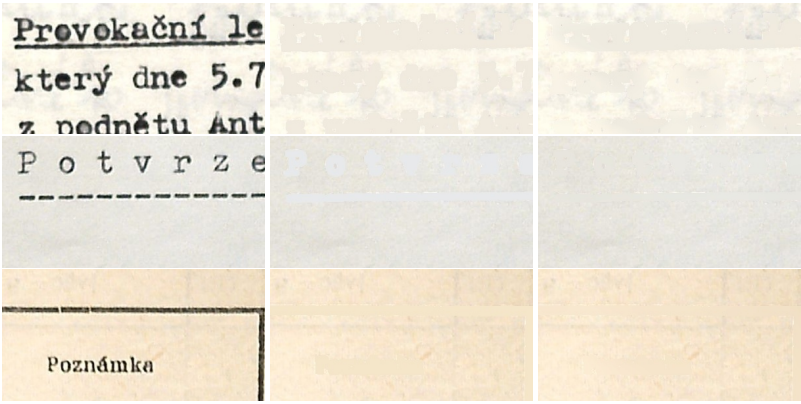

Fig. 1. Every step of the algorithm demonstrated on the tiles of input documents. The left column contains examples of input documents. The results after mean value replacement are presented in the middle column. The right column contains result after NEIGHBOR_MEAN_ADAPT – i.e., local adaptation of the mean color to eliminate "shadows" of erased characters

3. Background Generation. The task of synthetic image generation can be handled in different ways. This paper presents a method based on a neural network topology called autoencoder [3, 8]. This topology usually employs a bow-tie structure, where the first part encodes the input data into a latent space, and the second part then decodes the vector from latent space back to the shape of the input data. In the usual implementation of a plain

autoencoder, there are no restrictions applied to the latent space. Without any restrictions, the autoencoder acts as a memory with compression. To improve the idea of autoencoder into a generative model Variational Autoencoder was developed [12, 16, 23]. This approach forces the latent space to follow a Gaussian distribution. This typically leads to blurry images generated at the output of the VAE. We have discovered such behavior is to our advantage and developed the background generator for our synthetic data based on the VAE. The blur typical for the VAE introduced into the generated data helps us to improve the results of the Background Extraction algorithm (Section 2). The VAE eliminates the artifacts from the training set by compressing the data and then generating them back from the Gaussian latent space.

The training data for the VAE were obtained by Background Extraction algorithm in Section 2. The original scans of old documents were rid of the text by the background extraction algorithm, and 685 of them were used for training. The structure of the network is described in detail in Table 1. The latent space is created by two fully connected layers with size 250 neurons representing the mean and the variance of the Gaussian distribution with set values of zero mean and unit variance for each individual neuron. The training used the following parameters: learning rate of 0.001 with RMSprop optimizer over 1000 epochs. The input images were resized from the original resolution to the size of $128 \times 96 \times 3$. This size roughly represents the original ratio of the training data.

Only the decoder part of the VAE is used after training as a generator for synthetic backgrounds. The background is generated from an input vector of size 250. The values of the vector are sampled from the Gaussian distribution with zero mean and standard deviation of 0.15. This value was selected experimentally based on the analysis of the generated data. The images generated with a higher value of standard deviation produced various unrealistic backgrounds with different artifacts and colors which were not suitable for the task of synthesizing artificial images of old documents. Before the data can be processed by the decoder, there is a necessary step of resizing the single-dimensional vector of the input vector to a three-dimensional shape that can then be processed by deconvolutions. The needed shape is taken from the output of the last convolutional layer of the encoder. The output of the decoder is resized to the original resolution of the training data ($2480 \times 3504 \times 3$) using linear interpolation.

The main contribution of this approach is the ability to create a general background that is very similar to the real old paper. The algorithm is able to produce various colors from clean white paper, through gray paper, to yellow recycled paper typically used during older eras. The process removes typical

artifacts from the training data like water damage, coffee circle stains, and letters visible from the other side of the paper, which are typically present due to the nature of the typewriting. These may be added to the generated background if needed in our future research.

Table 1. Structure of VAE

Encoder	Decoder
64 conv(2×2), ReLU	deconv(3×3), ReLU
64 conv(2×2), BN, ReLU	deconv(3×3), ReLU
64 conv(3×3), ReLU	deconv(3×3), ReLU
64 conv(3×3), BN, ReLU	conv(2×2), sigmoid
500 fully-connected	

The encoder is composed of four convolutional layers with 64 kernels with the ReLU activation function. Every other convolutional layer is followed by batch normalization. The intermediate layer with 500 neurons has tanh activation function. The latent space is represented by two fully connected layers with 250 neurons and a linear activation function. The decoder mirrors the structure of the encoder.

4. Synthetic Image with Text Generation. The first part of creating a synthetic image is generating the background. The background image is created by our trained VAE model described in Section 3. The VAE neural network is fast enough to generate the images on the fly. The second part is adding synthetic text to the background. The font used to generate the synthetic text is Bohemian typewriter. A sample image with the font is depicted in Figure 2. We have used this font as it was corresponding to our real data from scanned typewritten documents. The font size was experimentally set to 45 pixels. This size is the closest to the real data, and it also depends on the resolution of the background. The font can be easily replaced by any other desirable font. This change does not impact the performance of our algorithm.

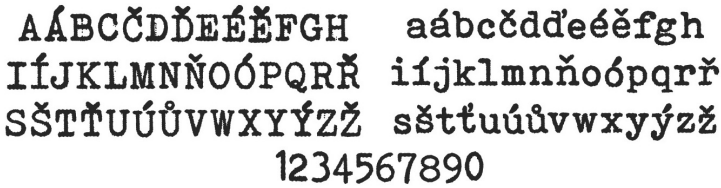

Fig. 2. Bohemian typewriter font used for generating synthetic documents

We have implemented an input from a text file to generate the text in the synthetic images. The text in the file can be composed of any UTF-8 characters. The contents of the text file are not limited by the output font. If the output font does not contain the character from the input file, it is replaced by a dummy placeholder.

The background is generated after the input file is processed. Our VAE neural network generator outputs an image with a resolution of 2480×3504. This resolution was chosen to correspond to the scanned documents in our dataset. The image has three color channels (RGB). Generated backgrounds are unique and unpredictable in the sense of color and noise. The general distribution of colors generated by our VAE network is given by the distribution in the training set. The pre-trained VAE model can be easily changed for another model if another type of background is needed.

The first instance of the text is generated on a white background with a resolution corresponding to the synthetic background (2480×3504). The white background is used to print out the characters with zero brightness. The text output is formatted into a predefined block. The block corresponds with a predefined area in the image where the text is supposed to be placed. These areas are easily added and/or modified to create a desired format of the output document, for example, two columns, two paragraphs, zig-zag, etc. The position of the area is given by the set of coordinates (x,y) that represents the top left corner. The second parameter of the area is its width and height of (w,h). The text is continuously printed into the predefined areas until all of them are filled, or there is no more text on the input. After that, if any text is remaining, it is omitted. A random offset is added to the coordinates (x,y) to add a more natural appearance to the synthetic data. Examples of the generated images are in Figures 3 and 4.

The position of every character is calculated with respect to its size. This position is further augmented with an offset generator. The offset can reach values of ± 3 pixels in both directions of x and y coordinates. This value is randomly added to each character. We have implemented this generator to simulate better the text produced by older types of typewriters where all the characters were not printed in a straight line. The offsets were selected experimentally to respect the size of the font and the resolution of the whole image. The value of the parameter can be easily changed if needed.

Various defects can be found in archived typewritten documents. One of them is a variance in the brightness of the old paper versus a new paper. We introduce a method for simulating this effect in our synthetic data. We generate a Gaussian noise $\mathcal{N}\left(\mu, \sigma^2\right)$ (where $\mu = 0$ and $\sigma^2 = 0.3$) and reshape it into the size width = $\frac{2480}{8}$ and height = $\frac{3504}{8}$. This noise matrix is then reshaped

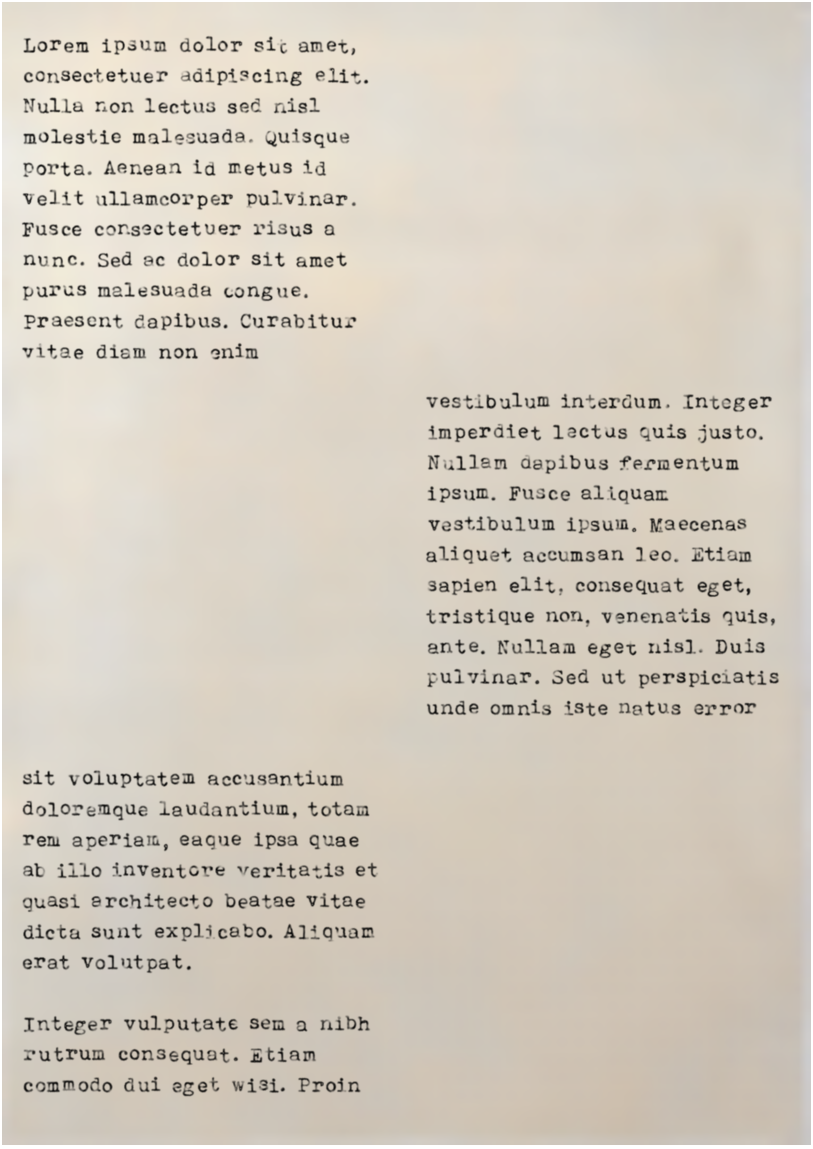

Fig. 3. Example of the layout areas: zig-zag text areas

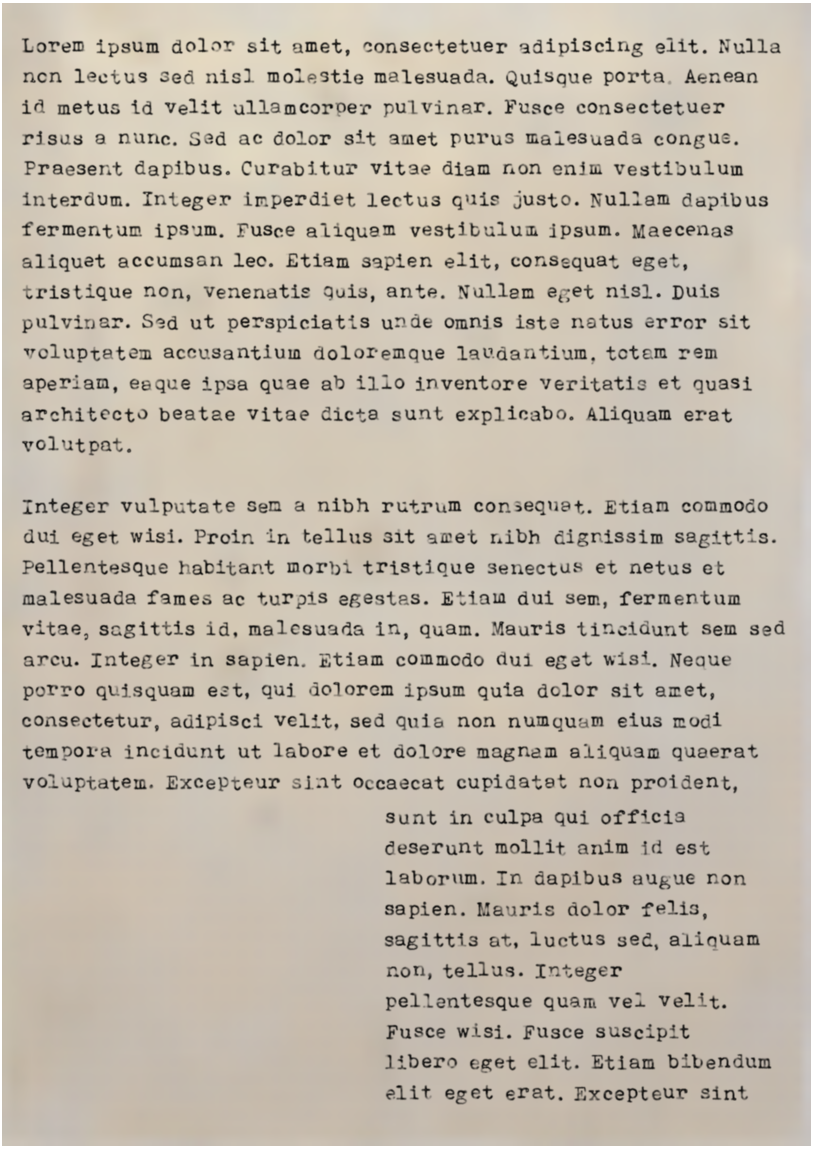

Fig. 4. Example of the layout areas: change of the width in the middle of the second paragraph

into the size of the synthetic image using linear interpolation. The noise is then added to the synthetic image (at the locations where the characters of the text are present). The final values are then clipped for the purpose of the consistency of the image data. The resulting image is further augmented by a blurring filter with a local averaging window of size 7×7. This process produces a synthetic image with blurred text areas. This image is then merged with the synthetic background generated by our VAE network. One last operation in the form of a blur filter is then applied to the whole image using a local averaging window of size 5×5.

We can create a bounding box from the knowledge of the parameters of each character (coordinates (x, y), character offset, and font size). The size of the bounding box is constant since the font width of the typewriter is not changing during writing. We have aimed for bounding boxes that are considered minimal. We have implemented a method to decrease the size of the bounding boxes. The synthetic image is transformed into grayscale at first. Then the contour of each character is extracted. Bounding boxes are then resized to touch the contours found in the previous step. The same has been done for page's, word's, and line's bounding boxes. Examples of character bounding boxes are shown in Figure 5.

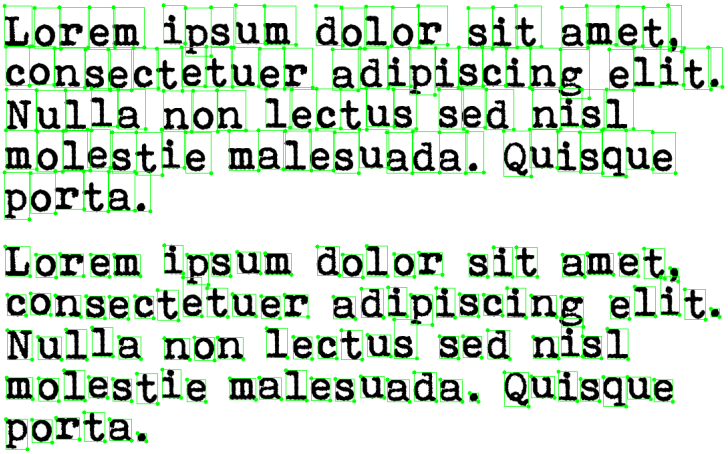

Fig. 5. Examples of character's bounding boxes: (top) the original bounding boxes, (bottom) fine–tuned character's bounding boxes

All collected data about images, words, characters, and lines are stored in files (in pickle data files and png images). This data can be utilized in further experiments and analysis.

The pickle data file for every page is structured in the following manner: the printing areas of a defined layout store the objects of lines. The line object contains a list of word objects, and each word object contains a list of character objects. The objects contain precise x and y global coordinates and the height and width of its bounding box. The file also contains information about the number of objects. We have also generated pixel-wise data with binary information about each character's pixel coordinates. This information was obtained by thresholding the white background image with the generated text before it was merged with the VAE background. The storage structure is depicted in Figure 6.

The data structure is designed to be universally usable in various tasks. We have used it for training several types of neural networks. It can also be used as ground truth data for benchmarking existing algorithms. A comparison of generated vs real text is depicted in Figure 7.

5. Experiments. To test the efficiency of the proposed method, we have designed an experiment to measure the accuracy of an OCR system on both the generated synthetic data and real scanned documents. We have chosen the Tesseract system for the OCR task. The hypothesis is that if the generated data have valid properties, Tesseract should perform similarly on the generated data as on the real data. For this purpose, we manually annotated 24 documents with a total of 25 292 valid characters. The formula to compute the accuracy was:

$$p = \left(1 - \frac{l}{m}\right) \cdot 100 \quad [\%], \tag{1}$$

where l is the Levenshtein distance between the two strings representing the OCR results of synthetic, respectively real data and m is the length of the longer string. The strings were rid of white characters, that do not bear any valuable information, such as spaces, newlines, dots, colons, etc.

Another metric we used was Word recall. In this case, we count the percentage of words from the ground truth text that we are able to recover with the OCR system. In the experiments, four datasets were tested. The first one is a dataset of real documents, and the other three are artificial datasets of documents generated by our proposed algorithms from the same text, as is in the real dataset. The artificial synthetic datasets were generated with various amount of character noise – i.e., damage in the structure of printed characters in the scan (see Section 4). The results are shown in Table 2. It can be seen that we are able to control the properties of the algorithm so that the results of the OCR system are close to the results from the real data. In Table 2 you can see Mean accuracy – i.e., character accuracy of OCR system, standard

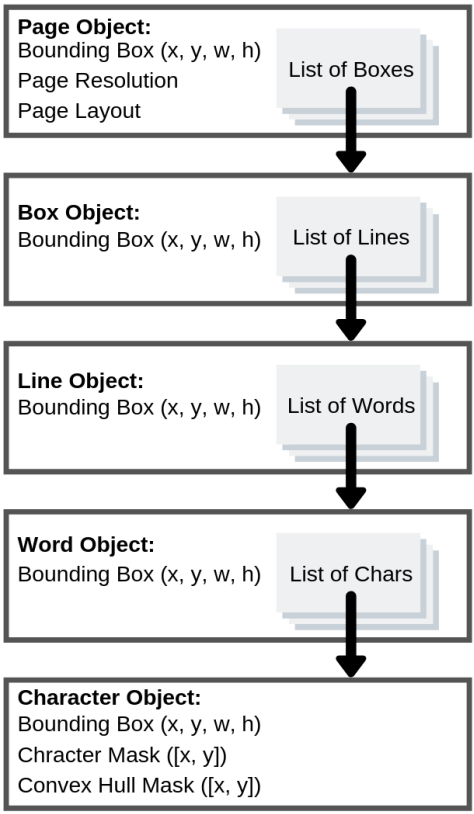

Fig. 6. Pickle data file structure hierarchy

deviation, and word recall. The word recall is important for measuring the performance of the OCR system for the purpose of information retrieval.

The results in Table 2 show that the dataset denoted Generated three is similarly challenging for the Tesseract system as the real data. Moreover, when observing the recognized texts, the OCR makes very similar mistakes as in the real scanned documents. The character noise used in this dataset can be used for generating benchmark datasets for comparing state of the art OCR systems. We can also control the amount of character noise to generate less challenging data. This attribute of the algorithm can be beneficial for training DNNs. First, a DNN can be trained on easier data, and then the difficulty of

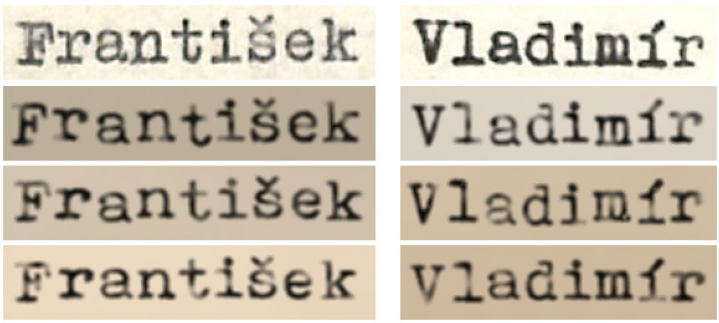

Fig. 7. Examples of generated text. (top row) original scanned text sample. (following rows) generated text

Table 2. OCR accuracy of real and generated scans. The generated datasets have different amount of noise applied: Generated 1 < Generated 2 < Generated 3

Dataset	Mean accuracy	STD	Word recall
Real data	0.73	0.14	0.79
Generated 1	0.94	0.08	0.90
Generated 2	0.92	0.10	0.83
Generated 3	0.88	0.12	0.69

the data can be progressively raised. One can also generate baseline datasets that should be recognized flawlessly.

6. Semantic Segmentation. Inspired by [2, 18, 20], we propose a method based on a Fully-Convolutional neural network with Encoder-Decoder structure, which proved itself to be perfectly suitable for semantic segmentation tasks. First, we designed a simple plain Encoder-Decoder network (our baseline architecture, for exact configuration, see Table 3), nevertheless, this setup did not provide satisfactory results. Therefore, we implemented a couple of improvements. To be more specific, our final neural network architecture is based on the work of Ronnenberger et al. [20] and their U-Net architecture (see Table 4). In comparison to standard Encoder-Decoder, this architecture utilizes skip connections between each layer i in the encoder and layer n-i in the decoder, where n is the total number of layers. The skip connections are utilized for element-wise addition of the activations from layer i and layer n-i. Experiments proved that usage of skip connections allowed better propagation of information through the network and, therefore, significantly improved

classification (or segmentation) results. In our future work, we would also like to test the concatenation of the activations instead of their summation.

A good practice before designing the architecture of a neural network is to understand the input data. In the case of visual processing, the receptive field is an important attribute of the individual layers. Since we want to read a text composed into lines, we need to figure out the average height of a text line so that the receptive field of the deepest layer is in compliance with this height. This approach is supported by the fact that text is highly dependent in one line and less dependent in the vertical direction. In the original resolution of 2480×3504 pixels, the average line-height is 36 pixels. Due to the memory limitations during training, the original resolution was reduced to 620×876, thus, the average height in the lower resolution is 9 pixels.

During the design of the neural network architecture, we utilize standard (de)convolution-batch normalization-ReLU sequence for all the convolutions and deconvolutions except the last one, which performs classification task and therefore uses classical Softmax activation function. All the convolutions and deconvolutions in this work are used with stride 1. We would also like to point out the total max-pooling operation omission, which stems from the optimal size of the convolutional receptive field considering the line average height.

Table 3. Structure of our baseline architecture

Encoder	**Decoder**
Conv(16, 3×3), BN, ReLU	Deconv(64, 5×5), BN, ReLU
Conv(32, 3×3), BN, ReLU	Deconv(32, 3×3), BN, ReLU
Conv(64, 5×5), BN, ReLU	Deconv(16, 3×3), BN, ReLU
	Conv(N, 1×1), Softmax

The encoder is composed of three convolutional layers with 16, 32, and 64 kernels, followed by batch normalization and ReLU activation function. The decoder mirrors this structure. N in the last convolutional layer of the decoder is the number of classes.

Table 4. Structure of our final architecture

Encoder	**Decoder**
Conv1(16, 3×3), BN, ReLU	Deconv1(64, 5×5), BN, ReLU
Conv2(32, 3×3), BN, ReLU	Deconv2(32, 3×3), BN, ReLU
Conv3(32, 3×3), BN, ReLU	Deconv3(32, 3×3), BN, ReLU
Conv4(64, 3×3), BN, ReLU	Deconv4(16, 3×3), BN, ReLU
	ConvF(N, 1×1), Softmax

The encoder is composed of four convolutional layers with 16, 32, 64, and 64 kernels, followed by batch normalization and ReLU activation function again. The decoder mirrors this structure. *N* in the last convolutional layer ConvF of the decoder is the number of classes. There is a skip connection between each layer *i* and layer *n-i*.

All tested neural network architectures were implemented in Python using Chainer deep learning framework [1, 22]. Also, in all experiments, we use the cross-entropy loss for the network training. In the following subsections, we describe the experimental setting and the results of two experiments. For both of them, we are providing quantitative results for testing synthetic data and qualitative results for both synthetic and real data.

6.1. Two-class classification. In the first experiment, we train baseline architecture to perform semantic segmentation of input image into two classes: text, and background, i.e., last convolutional layer has two kernels – one for each class and the softmax activation function is used to obtain a probability measure in each location. To further elaborate, for each class, the network produces one output map with segmentation for this specific class. The final segmentation map can be constructed by combining the results from all output maps in a simple manner.

Standard mini-batch SGD optimization with mini-batch size 2 – memory limitation – is used in our architecture. We used the learning rate with step decay. In particular, the learning rate started on value 0.01, and every 10 epochs, the step decay 0.1 was applied. The recognition accuracy of the network on the testing set is 99.28% after 30 epochs. An example of a result of a synthetic document segmentation is shown in Figure 8.

Our dataset contains 150 thousand synthetic documents divided into three sets – i.e., training, development, and testing set. The total number is split as follows. The training set contains 100 thousand documents, the development set 20 thousand documents, and finally testing set contains 30 thousand documents.

In Figure 9, an example of real document segmentation is shown. It can be noted that the network is capable of filtering out the document background. Moreover, it even removes paper inaccuracies and other defects. On the other hand, the network leaves parts of the text which are not wanted in our result. In the example, the underline under characters is unwanted because it can influence the accuracy of the OCR algorithm. The reason for this phenomenon is caused by the absence of this type of distraction in the synthetic data. Nevertheless, we argue that this approach can be used as a part of a pre-processing pipeline for the standard OCR algorithm and hopefully improves its final results.

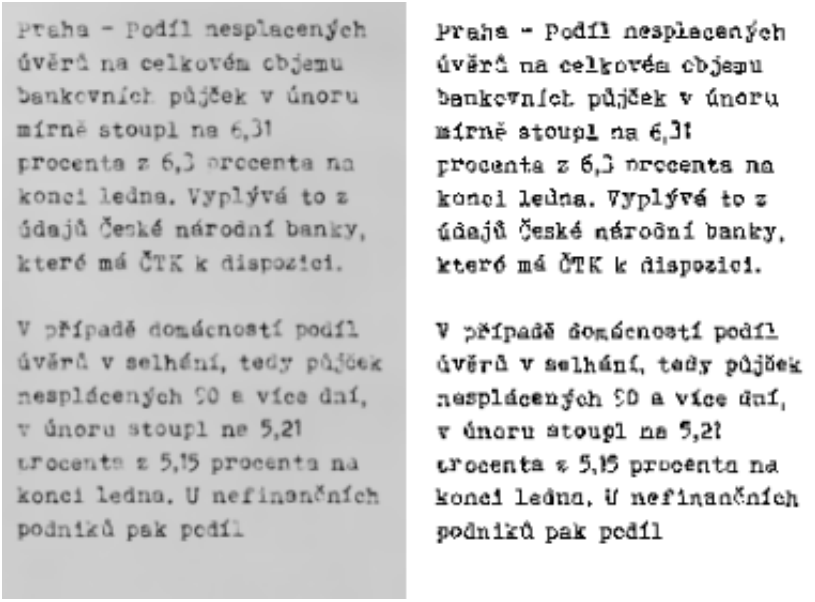

Fig. 8. A result (on the right) of the semantic segmentation for the class *text* of the synthetic document (on the left)

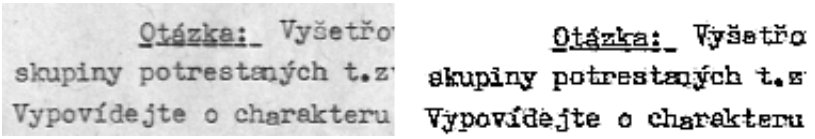

Fig. 9. A result (on the right) of the semantic segmentation for the class *text* of the real historical document (on the left)

Segmentation of convex hulls instead of direct character segmentation is used in our research. The idea was proposed in the paper Hajic et al. [11]. The paper contains a statement that the training of the network should be better with a convex hull approach. The idea is verified on the task of musical symbol recognition. In Figure 10, an example of convex hulls of letters a, b, and c are shown.

To try convex hulls on our segmentation problem, we had to regenerate synthetic data labels. We trained the network using the same training settings to predict hulls instead of characters. The recognition accuracy was improved by 0.27% on the test set using this approach. It represents an improvement of 37.50% w.r.t. the approach based on the direct segmentation of characters. On

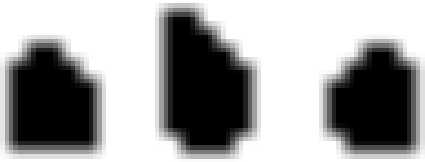

Fig. 10. Examples of convex hulls for letters *a*, *b*, and *c*

the other hand, it does not solve any of the systematic problems present in the first method.

Table 5. Baseline architecture results of per-pixel recognition accuracy of two-class segmentation on the synthetic documents

NN architecture	Development set	Test set
Direct-char	99.32%	99.28%
Convex-hull	99.60%	99.55%

Generation of the problematic unwanted parts of the text is one of the tasks that we want to address in our future research. Their presence in the training sets should improve the results of the neural network.

6.2. Single-character classification. The second experiment is focused on the semantic segmentation into 107 classes – i.e., classes of Czech language text characters plus one for the background. This task is called a single-character classification. We assume that it can avoid problems of two-class segmentation. During the experiment, we train the network using both direct-character and convex hull segmentation. The recognition accuracy of the network reached 97.75%, 97.58%, respectively.

Despite the good quality of results, we find out that the network provides almost perfect prediction for frequent characters – e.g. `a` and `e` – but significantly worse results for the rare ones – e.g. `F` and `G`. The reason for this fact is the nature of the Czech language because some of the characters are rare. Consistently, there are a small number of occurrences of these characters in the training data.

There are basically two possible solutions to overcome this flaw. First, we can regenerate the training set with a uniform distribution of the characters. This approach will definitely solve the problem for the synthetic data, but it will likely provide worse results for the real documents, which meet the unbalanced frequency of characters of the Czech language.

Table 6. Comparison per-pixel recognition accuracy of single-character segmentation on the synthetic document test set for the baseline and the final architecture

NN architecture	Direct-char	Convex-hull
Baseline	97.75%	97.58%
Baseline_weighted	97.83%	97.75%
Final	99.14%	99.17%
Final_weighted	99.53%	99.52%

The second option is that we preserve the same training set, but we motivate our network to stop ignoring the rare classes. Therefore, we weight the loss from individual classes w.r.t. their frequency, i.e., the loss from the less frequent classes introduces a larger penalty than from the frequent ones. The frequencies of individual characters were obtained from 10 thousand real internet news documents. Weighted categorical cross-entropy slightly improves results. However, the obtained results still were not satisfying. We tested a few modifications to our baseline neural network architecture. In Table 6, the results of the baseline and our final architecture can be found. The results while using standard categorical cross-entropy are listed too. We would like to point out the fact that convex-hull prediction reached only comparable results despite that in theory, it should significantly improve them. In our opinion, it is caused by the relative similarity of the characters' convex hull and the characters themselves as opposed to the findings in the work [11] with convex hulls of musical symbols. We are also providing qualitative results (see Figure 11 and Figure 12) of the final architecture using weighted cross-entropy loss.

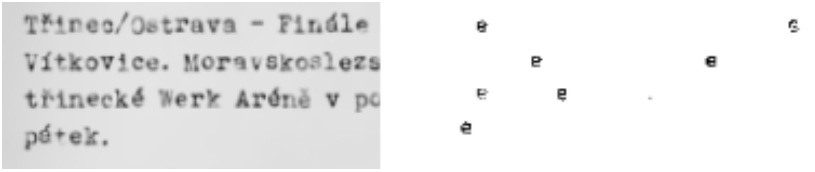

Fig. 11. A result (on the right) of the semantic segmentation for letter *e* of a synthetic document (on the left) using the final architecture using weighted cross-entropy loss

The final network architecture and training process reached good accuracy. Therefore, we want to use it as a pre-processing part for a novel OCR algorithm for scanned historical documents in our future research. On the other hand, there are still problems with the segmentation of rare characters that should be addressed. There is also some noise caused by the appearance of

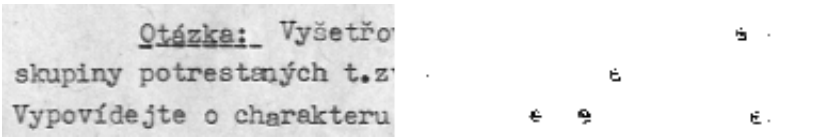

Fig. 12. A result (on the right) of the semantic segmentation for letter *e* of the real historical document (on the left) using the final architecture using weighted cross-entropy loss

scanned documents. In the future, we want to try more complex architectures together with an extended training set to solve these problems. Moreover, the parallelization of the training process can increase the batch size and lead to results of better quality.

7. Conclusion and Future Work. We have presented an algorithm for generating synthetic images resembling real-world scanned typewritten documents. The algorithm is highly modifiable in terms of generating different classes of documents. The algorithm is composed of modules handling smaller tasks. They can be easily switched and modified, which also enables appearance tweaks for any given class of documents. The modules include a dynamic generation of backgrounds using VAE, different standard fonts, and layout generation. The attributes of the printed text can be easily interchanged. There is the possibility of location noise addition, blurring of the printed text, or the whole generated image. We test the properties of the generated documents, and we observe that through the lens of an existing OCR system, they are very similar to real-world scanned documents. The generated data can be used to train, test, and evaluate new or existing OCR algorithms. In this work, we utilize the algorithm to generate vast amounts of training data for our semantic segmentation approach of text detection and character classification. We use an approach called semantic segmentation, which involves a fully-convolutional neural network.

Two different tests were performed: two-class classification and character classification. With the best training settings and final fully-convolutional encoder-decoder architecture inspired by U-Net, we reached 99.28% recognition accuracy, 99.52% respectively, for the synthetic data. We also proposed segmentation of convex hulls instead of direct segmentation of characters, which, unfortunately, did not provide any legible improvement. We also provide qualitative results for real data. For our future research, we have several plans. First, we would like to extend our generator with options to generate new text features like text underlining or line margins. Second, we plan to use two-class classification results as a part of a pre-processing pipeline for the standard

OCR algorithm. Third, we will employ more complex architecture in the character classification task. Last, but not least, we will develop a text decoder to be able to compare reached results with other OCR algorithms.

References

1. Akiba T., Fukuda K., Suzuki S. ChainerMN: Scalable distributed deep learning framework. arXiv preprint arXiv:1710.11351. 2017.
2. Badrinarayanan V., Kendall A., Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. vol. 39(12). pp. 2481–2495.
3. Bengio Y. et al. Learning deep architectures for AI. Foundations and trends® in Machine Learning. 2009. vol. 2(1). pp. 1–127.
4. Breuel T. Recent progress on the OCRopus OCR system. Proceedings of the International Workshop on Multilingual OCR. 2009. pp. 2.
5. Bureš L., Neduchal P., Hlavác M., Hrúz M. Generation of synthetic images of full-text documents. International Conference on Speech and Computer. 2018. pp. 68–75.
6. Chen L.C., Papandreou G., Schroff F., Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. 2017.
7. Chernyshova Y.S., Gayer A.V., Sheshkus A.V. Generation method of synthetic training data for mobile OCR system. Tenth International Conference on Machine Vision (ICMV 2017). 2018. vol. 10696. pp. 106962G.
8. Dumas T., Roumy A., Guillemot C. Autoencoder based image compression: can the learning be quantization independent? 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018. pp. 1188–1192.
9. Gruber I., Hlavác M., Hrúz M., Železný M. Semantic segmentation of historical documents via fully-convolutional neural network. International Conference on Speech and Computer. 2019. pp. 142–149.
10. Gupta A., Vedaldi A., Zisserman A. Synthetic data for text localization in natural images. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 2315–2324.
11. Hajic J., Dorfer M., Widmer G., Pecina P. Towards full-pipeline handwritten OMR with musical symbol detection by u-nets. ISMIR. 2018. pp. 225–232.
12. Huang H., He R., Sun Z., Tan T. Introvae: Introspective variational autoencoders for photographic image synthesis. Advances in Neural Information Processing Systems. 2018. pp. 52–63.
13. Huang W., Qiao Y., Tang X. Robust scene text detection with convolution neural network induced mser trees. European Conference on Computer Vision. 2014. pp. 497–511.
14. Jaderberg M., Vedaldi A., Zisserman A. Deep features for text spotting. European Conference on Computer Vision. 2014. pp. 512–528.
15. Jaderberg M., Simonyan K., Vedaldi A., Zisserman A. Reading text in the wild with convolutional neural networks. International Journal of Computer Vision. 2016. vol. 116(1). pp. 1–20.
16. Kingma D.P., Welling M. Auto-encoding variational bayes. International Conference on Learning Representations. 2014. 21 p.
17. Lin G. et al. Refinenet: Multi-path refinement networks for dense prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019.
18. Noh H., Hong S., Han B. Learning deconvolution network for semantic segmentation. Proceedings of the IEEE International conference on computer vision. 2015. pp. 1520–1528.

19. Otsu N. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics. 1979. vol. 9(1). pp. 62–66.
20. Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. 2015. pp. 234–241.
21. Smith R. An overview of the tesseract OCR engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). 2007. vol. 2. pp. 629–633.
22. Tokui S., Oono K., Hido S., Clayton J. Chainer: a next-generation open source framework for deep learning. Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS). 2015. vol. 5. pp. 1–6.
23. Wen S. et al. Variational autoencoder based image compression with pyramidal features and context entropy model. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019. pp. 0–0.
24. Zhao H. et al. Pyramid scene parsing network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. pp. 2881–2890.
25. Zhou X. et al. EAST: An efficient and accurate scene text detector. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. pp. 5551–5560.

Lukáš Bureš — Ph.D. Student, New Technologies for the Information Society (Research centre of Faculty of Applied Sciences), University of West Bohemia. Research interests: machine learning, computer vision, visual keypoint detection and description. The number of publications — 1. lbures@ntis.zcu.cz; 8, Technická, 306 14, Plzen, Czechia; office phone: +420 377 632 145.

Ivan Gruber — Ph.D. Student, NTIS - New Technologies for the Information Society (Research Centre of Faculty of Applied Sciences), University of West Bohemia. Research interests: machine learning, face recognition, image recognition. The number of publications — 14. grubiv@ntis.zcu.cz; 8, Technická, 306 14, Plzen, Czech Republic; office phone: +420 377 632 137.

Petr Neduchal — Ph.D. Student, NTIS - New Technologies for the Information Society (Research Centre of Faculty of Applied Sciences), University of West Bohemia. Research interests: mobile robotics, simultaneous localization and mapping, robot exploration, computer vision, machine learning. The number of publications — 7. neduchal@kky.zcu.cz; 8, Technická, 306 14, Plzen, Czech Republic; office phone: +420 377 632 145.

Miroslav Hlaváč — Ph.D. Student, NTIS - New Technologies for the Information Society (Research Centre of Faculty of Applied Sciences), University of West Bohemia. Research interests: machine learning, image recognition, lip reading, audiovisual speech recognition. The number of publications — 1. mhlavac@ntis.zcu.cz; 8, Technická, 306 14, Plzen, Czech Republic; office phone: +420 377 632 136.

Marek Hrúz — Ph.D., Senior Researcher, NTIS - New Technologies for the Information Society (Research Centre of Faculty of Applied Sciences), University of West Bohemia. Research interests: computer vision, machine learning, deep learning, data mining, multi-modal data processing. The number of publications — 42. mhruz@ntis.zcu.cz; 8, Technická, 306 14, Plzen, Czech Republic; office phone: +420 377 632 555.

Acknowledgements. This work was supported by the Ministry of Education of the Czech Republic, project No. LTARF18017 and Ministry of Education, Youth and Sports of the Czech Republic project No. LO1506. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the

programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated. The work has been supported by the grant of the University of West Bohemia, project No. SGS-2019-027.

УДК 004.9 **DOI** 10.15622/sp.2019.18.6.1381-1406

Л. Буреш, И. Грубер, П. Недухал, М. Главач, М. Груз

СЕГМЕНТАЦИЯ СЕМАНТИЧЕСКОГО ТЕКСТА ПО ИСКУССТВЕННОМУ ИЗОБРАЖЕНИЮ ПОЛНОТЕКСТОВЫХ ДОКУМЕНТОВ

Буреш Л., Грубер И., Недухал П., Главач М., Груз М. **Сегментация семантического текста по искусственному изображению полнотекстовых документов.**

Аннотация. Предлагается разделенный на несколько модулей алгоритм для создания изображений полнотекстовых документов. Эти изображения можно использовать для обучения, тестирования и оценки моделей оптического распознавания символов (ОПР). Алгоритм является модульным, отдельные части могут быть изменены и настроены для создания желаемых изображений. Описывается метод получения фоновых изображений бумаги из уже оцифрованных документов. Для этого используется новый, основанный на вариационном автоэнкодере подход к обучению генеративной модели. Эти фоны позволяют сразу же сгенерировать такие же фоновые изображения, как те, на которых производилось обучение.

Для получения правдоподобного эффекта старения в модуле печати текста используются большие текстовые блоки, типы шрифтов и вариативность изменения яркости символов.

Поддерживаются несколько типов макетов страницы. Система генерирует подробную структурированную аннотацию искусственного изображения. Для сравнения реальных изображений с искусственно созданными используется программа Тессеракт ОПР. Точность распознавания приблизительно схожа, что указывает на правильность сгенерированных искусственных изображений. Более того, допущенные системой ОПР ошибки в обоих случаях очень похожи. На основе сгенерированных изображений была обучена архитектура сверточная кодер-декодер нейронная сеть полностью для семантической сегментации отдельных символов. Благодаря этой архитектуре достигнута точность распознавания 99,28% в тестовом наборе синтетических документов.

Ключевые слова: генерация искусственных изображений, сегментация семантического текста, вариационный автоэнкодер, OCR, оптическое распознавание символов, распознавание текста, генерация искусственно состаренного текста.

Буреш Лукаш — аспирант, НТИО - Новые технологии для информационного общества (исследовательский центр факультета прикладных наук), Западно-чешский университет. Область научных интересов: машинное обучение, компьютерное зрение, визуальное обнаружение и описание ключевых точек. Число научных публикаций — 1. lbures@ntis.zcu.cz; ул. Техническая, 8, 306 14, Пльзень, Чехия; р.т.: +420 377 632 145.

Грубер Иван — аспирант, НТИО - Новые технологии для информационного общества (исследовательский центр факультета прикладных наук), Западно-чешский университет. Область научных интересов: машинное обучение, распознавание лиц, распознавание изображений. Число научных публикаций — 14. grubiv@ntis.zcu.cz; ул. Техническая, 8, 306 14, Пльзень, Чехия; р.т.: +420 377 632 137.

Недухал Петр — аспирант, НТИО - Новые технологии для информационного общества (исследовательский центр факультета прикладных наук), Западно-чешский университет. Область научных интересов: мобильная робототехника, одновременная локализация и картирование, исследование робота, компьютерное зрение, машинное обучение. Число

научных публикаций — 7. neduchal@kky.zcu.cz; ул. Техническая, 8, 306 14, Пльзень, Чехия; р.т.: +420 377 632 145.

Главач Мирослав — аспирант, НТИО - Новые технологии для информационного общества (исследовательский центр факультета прикладных наук), Западно-чешский университет. Область научных интересов: машинное обучение, компьютерное зрение, лабиомансия, аудиовизуальное распознавание речи. Число научных публикаций — 1. mhlavac@ntis.zcu.cz; ул. Техническая, 8, 306 14, Пльзень, Чехия; р.т.: +420 377 632 136.

Груз Марек — Ph.D., старший научный сотрудник, НТИО - Новые технологии для информационного общества (исследовательский центр факультета прикладных наук), Западно-чешский университет. Область научных интересов: компьютерное зрение, машинное обучение, глубинное обучение, интеллектуальный анализ данных, мультимодальная обработка данных. Число научных публикаций — 42. mhruz@ntis.zcu.cz; ул. Техническая, 8, 306 14, Пльзень, Чехия; р.т.: +420 377 632 555.

Поддержка исследований. Данное исследование проведено при поддержке Министерства образования Чешской Республики (проект № LTARF18017) и Министерства образования, молодежи и спорта Чешской Республики (проект № LO1506). Также благодарим за предоставление доступа к хранилищам, принадлежащим участникам создания метацентра национальной грид-инфраструктуры в рамках программы «Проекты крупных инфраструктур для исследований, разработок и инноваций» (CESNET LM2015042). Работа выполнена при поддержке гранта Западно-чешского университета (проект № SGS-2019-027).

Литература

1. *Akiba T., Fukuda K., Suzuki S.* ChainerMN: Scalable distributed deep learning framework // arXiv preprint arXiv:1710.11351. 2017.
2. *Badrinarayanan V., Kendall A., Cipolla R.* Segnet: A deep convolutional encoder-decoder architecture for image segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. vol. 39(12). pp. 2481–2495.
3. *Bengio Y. et al.* Learning deep architectures for AI // Foundations and trends® in Machine Learning. 2009. vol. 2(1). pp. 1–127.
4. *Breuel T.* Recent progress on the OCRopus OCR system // Proceedings of the Inter-national Workshop on Multilingual OCR. 2009. pp. 2.
5. *Bureš L., Neduchal P., Hlavác M., Hrúz M.* Generation of synthetic images of full-text documents // International Conference on Speech and Computer. 2018. pp. 68–75.
6. *Chen L.C., Papandreou G., Schroff F., Adam H.* Rethinking atrous convolution for semantic image segmentation // arXiv preprint arXiv:1706.05587. 2017.
7. *Chernyshova Y.S., Gayer A.V., Sheshkus A.V.* Generation method of synthetic training data for mobile OCR system // Tenth International Conference on Machine Vision (ICMV 2017). 2018. vol. 10696. pp. 106962G.
8. *Dumas T., Roumy A., Guillemot C.* Autoencoder based image compression: can the learning be quantization independent? // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018. pp. 1188–1192.
9. *Gruber I., Hlavác M., Hrúz M., Železný M.* Semantic segmentation of historical documents via fully-convolutional neural network // International Conference on Speech and Computer. 2019. pp. 142–149.
10. *Gupta A., Vedaldi A., Zisserman A.* Synthetic data for text localization in natural images // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 2315–2324.
11. *Hajic J., Dorfer M., Widmer G., Pecina P.* Towards full-pipeline handwritten OMR with musical symbol detection by u-nets // ISMIR. 2018. pp. 225–232.

12. *Huang H., He R., Sun Z., Tan T.* Introvae: Introspective variational autoencoders for photographic image synthesis // Advances in Neural Information Processing Systems. 2018. pp. 52–63.
13. *Huang W., Qiao Y., Tang X.* Robust scene text detection with convolution neural network induced mser trees // European Conference on Computer Vision. 2014. pp. 497–511.
14. *Jaderberg M., Vedaldi A., Zisserman A.* Deep features for text spotting // European Conference on Computer Vision. 2014. pp. 512–528.
15. *Jaderberg M., Simonyan K., Vedaldi A., Zisserman A.* Reading text in the wild with convolutional neural networks // International Journal of Computer Vision. 2016. vol. 116(1). pp. 1–20.
16. *Kingma D.P., Welling M.* Auto-encoding variational bayes // International Conference on Learning Representations. 2014. 21 p.
17. *Lin G. et al.* Refinenet: Multi-path refinement networks for dense prediction // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019.
18. *Noh H., Hong S., Han B.* Learning deconvolution network for semantic segmentation // Proceedings of the IEEE International conference on computer vision. 2015. pp. 1520–1528.
19. *Otsu N.* A threshold selection method from gray-level histograms // IEEE transactions on systems, man, and cybernetics. 1979. vol. 9(1). pp. 62–66.
20. *Ronneberger O., Fischer P., Brox T.* U-net: Convolutional networks for biomedical image segmentation // International Conference on Medical image computing and computer-assisted intervention. 2015. pp. 234–241.
21. *Smith R.* An overview of the tesseract OCR engine // Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). 2007. vol. 2. pp. 629–633.
22. *Tokui S., Oono K., Hido S., Clayton J.* Chainer: a next-generation open source framework for deep learning // Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS). 2015. vol. 5. pp. 1–6.
23. *Wen S. et al.* Variational autoencoder based image compression with pyramidal features and context entropy model // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019. pp. 0–0.
24. *Zhao H. et al.* Pyramid scene parsing network // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. pp. 2881–2890.
25. *Zhou X. et al.* EAST: An efficient and accurate scene text detector // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. pp. 5551–5560.