Optimizing Algorithms of Text Detection on Images for Mobile Device using Structural Approaches

Vasyl Tereshchenko and Yaroslav Tereshchcenko

Taras Shevchenko National University of Kyiv, Addres 64/13, Volodymyrska Street, Kyiv, 01601, Ukraine

Abstract

The work is devoted to developing efficient algorithms for detecting text in an image obtained from mobile device's camera. The peculiarity of such algorithms is essential limitations of memory consumption and execution time (in our case: 5MB, ~1 s) while supporting detection quality on the level that cloud-based services provide. To ensure the efficiency of algorithms with specified constraints we propose a mixed model for detecting texts in images that involves preprocessing, detection, and post-processing. At the preprocessing stage the optimized SWT Voronoi diagram improving and expanding the features space for further text recognition processes an input RGB image. Cropped CNN solves detection under the specified limitations. Resulting false blocks having no text are discarded and separated parts with character sequences are combined into words and sentences using a filtering block that takes into account text style similarities. This approach allowed us to develop and test an implementation of algorithms for text detection in images for Android platform fitting mentioned memory footprint and execution time constraints.

Keywords [1]

Text detection, mobile devices, machine learning, convolution neural network, Voronoi diagram, preprocessing

1. Introduction

Text recognition on images with arbitrary backgrounds belongs to one of the actively developed problems of the computer vision. Despite the fact that methods for recognizing texts have been evolving for several decades they still have much space for further improvements. In particular in case of complex and heterogeneous scenes when a priory criteria for distinguishing a text from its background are unclear [1-7]. Such problems, for instance, are inherent for applications developed for mobile devices where there are tight limits on the runtime performance and the memory footprint. In turn, this motivates searching new optimal pipelines for balancing between text detection and its further processing (e.g. handwriting, print, style, and language recognition). It is worth noting that the localization of text blocks is not an easy task under variety of conditions like lighting, arrangement of the text in relation to the camera, presence of non-textual symbolic artifacts and graphic information along with text on the images, other distortions of the text. Examples of the text areas are inscriptions on billboards, buildings, institutions, road signs, participants of the movement (pedestrians and cars), as well as the text on the board, lecture notes, and other textual information (Fig. 1).

There are many approaches to solving this problem: methods based on correlation, contour and texture segmentation, discrete Fourier transformation, wavelet transformation, neural networks. All popular approaches conventionally belong to the following main classes: structural and based on machine learning (CNN-convolutional neural network) and combined (or mixed). Structural approaches include methods that take into account geometry and topology of image elements. In particular, these methods are based on the processing of images and the construction of data structures, as well as the use of image decomposition. For example, the most frequently used data

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: vtereshch@gmail.com (A. 1); y_ter@ukr.net (A. 2)
ORCID: 0000-0002-0139-6049 (A. 1); 0000-0002-8451-7634 (A. 2)

structures are lists, PHP, trees (k-trees, BSP trees, BD trees, BBD trees, QBD trees, VP trees, concatenable queue).

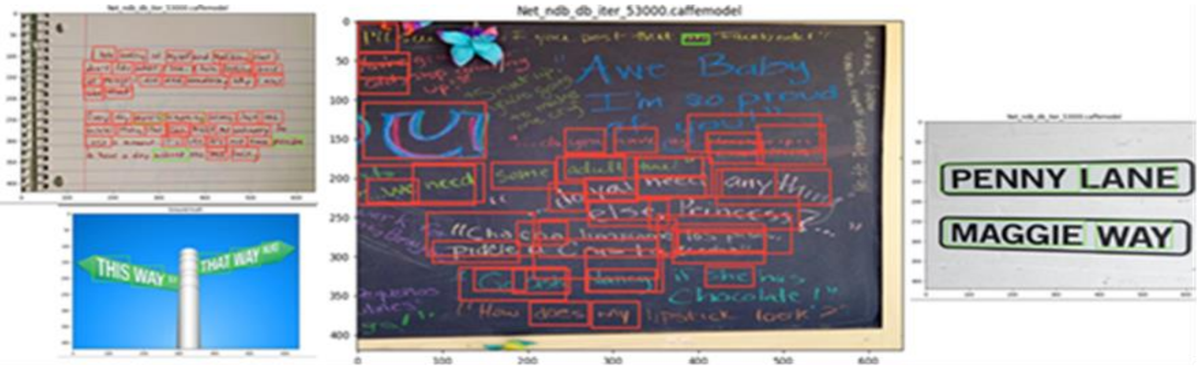


Figure 1: Examples of input images

We also use such decomposition tools as triangulation (Delaunay triangulation), Voronoi diagram, orthogonal recursive partition, and so on. Structural methods allow to reduce noises significantly at the preprocessing stage. It significantly improves quality and speed of detecting desired elements in the image. To date, there are many structural algorithms of automatic segmentation, which in general can be divided into two groups: the division into homogeneous regions [8 -12] and the allocation of contours [10,13-15]. There are many methods however none of them are universal enough.

Authors in [3] predict that the letters and words in the image, as a rule, have a constant (or width of the contour in a certain narrow range) the thickness of the stroke. Therefore, in their view, to identify such objects, it is promising to use the algorithm SWT (Stroke Width Transform). The width of the stroke, we can use not only as one of the features of the classification of areas, but also as a feature when combining areas in the words. Character contours within the described approach can be defined, for example, with the help of Canny boundary detector. However, it should be borne in mind that the SWT algorithm requires certain additional computational resources to combat errors and some other specific effects.

The next approach is related to the use of convolutional neural networks (CNN) to detect a text in an image. In particular, in [2] authors describe the automatic generation of features that will be used for recognition. It is proposed to create such features by means of machine learning, and as dataset for training take artificial images 8x8 pixels that contain fragments of text characters. To search in real images it will be enough to calculate the found features in the desired areas of the image. In papers [16-23] for the detection of a text on the image, the authors suggest using as a classifier the neural network (CNN), which, in comparison with classical neural networks has the following advantages: the ability to account for the spatial structure, reduction of architecture complexity and training, resistance to distortion of symbols. In our view, methods that use semantic knowledge about objects and algorithms of machine learning to improve the results of segmentation [21] are quite promising. The most suitable, in our opinion, is SSD and YOLO networks. These networks give good results to text detection using insignificant resources. However, we consider that it is not appropriate to use only neural networks in the way that we have to process a huge amount of redundant and unnecessary information, which results in loss of time and resources and thus restricts the ability to use them for mobile devices.

One of the main problems that we have to solve when implementing the proposed method is that the quality of the classification stage (including the use of convolutional neural networks) depends significantly on the volume and the representation of the training sample. Today, as datasets, we can use specially selected image bases, such as the ICDAR artificial intelligence algorithms database [22]. It should be noted that the number and type of images included in the training sample involved in the classifier's training determine the learning speed and the accuracy of the classification.

Modern approaches to solving the problem of text detection on a natural image usually consist of sequential use of algorithms, where the result of the work of the previous one is passed to the input of the next. Everything begins with the detection of low-level characters or strokes, after that the following steps are usually performed: filtering not-text components, constructing text strings and checking them. The presence of such a number of steps complicates the algorithm itself and reduces its reliability and flexibility. The accuracy of this approach depends on the accuracy of detection symbols methods, connected-components or sliding-window method. These methods generally examine the low-level features that were obtained when using SWT [3, 24], MSER [25, 26] or HOG [27] to distinguish text from the background. However, this does not guarantee the reliability of this approach, since individual strokes or symbols are identifying without context. For example, it's easier for a person to recognize a sequence of characters than a single letter, especially when it is implicit. These restrictions cause false detection of the text at the stage of letters detection, which in turn complicates the elimination of these errors. Moreover, the number of errors is accumulated with each stage of the algorithm [27]. In our work, we focus on one of the most efficient structural methods of text recognition on a natural image -SWT [3]. In particular, we propose an optimization and adaptation of this method for developing applications for mobile devices.

2. The Peculiarities of the Method SWT (Stroke-Width Transform) in the Text Detection Problems on the Image

The main idea of the SWT method is detecting gestures (strokes) of the equal width on binarized image (for example, obtained using the Canny algorithm [15]) that are likely candidates for the characters or letters of the text on a natural image. The method differs from other approaches in that it does not look for separated features per pixel, such as gradient or color. In addition, the SWT method does not use any language-related filtering mechanisms, such as statistics of gradient directions in the candidate window relating to a particular alphabet. This allows us to use this approach for multilingual text detection. Also, the method does not focus on the exact definition of the gesture (stroke) width, and on the trend detection that the stroke is an element of the text within a certain width of the contour. The next step of the algorithm is to group pixels into the list of candidates for the letters. Two adjacent pixels can be grouped if they have the same contour width. Then there is a filtering based on the mean square deviation, size, and other structural features. In the following, grouping words into blocks is by clustering methods.

In most cases, SWT gives better results in time than other algorithms. However, it can spend a lot of time to areas where there is no text due to a certain noise, or if the plot also has a uniform contour. Another drawback of SWT is its unreliability in cases where the width of stroke changes sufficiently. It can often happen in handwritten texts, or in certain font types of printed text. Therefore, we offer an algorithm for optimizing the procedure for detecting thickness and grouping letters into text blocks based on the use of the Voronoi diagram. It greatly speeds up the work of the SWT itself and thus can be applied to detect text on the image using mobile devices and special robot-technical devices.

Usually, the use of the Voronoi diagram is not new for recognition tasks, and in particular, a lot of works are devoted to the localization of a text in an image. Thus, N. C. Kha and N. Masaki [31, 32] offer a method for segmenting characters of handwritten text pages for the Japanese language. Using the Voronoi diagram, the authors divide text elements for qualitative using SWT. However, this modification does not apply to the main procedure of the SWT operator: the finding of the equal width strokes and candidates for characters or letters of the text. In [33], the authors propose a new approach to skeletonization of text images using the Voronoi diagram.

3. Model and Methods for Solving the Problem

In the methods for text localization on the image with limitations, much attention is focused on the execution time and memory required to store and implement algorithms. Text recognizing problem is one of the topical issues of computer vision. Despite the fact that the development of methods for recognizing texts is for several decades, this problem is far from completing for real images, which have complex and heterogeneous backgrounds and the absence of clear criteria for distinguishing text from background [1-7]. The problem is particularly acute in the development of applications for mobile and special robotic devices, where there are severe limitations on runtime and memory. This, in turn, leads to the development of new approaches and models for solving problems of text detection in the image and its further processing (handwriting, print, style, and speech recognition). For text localization with limitations, we pay attention to the runtime and memory needed to store and implement algorithms. Restrictions, for example, may be due to the adaptation of algorithms on mobile devices under Android. Therefore, to achieve efficiency of algorithms within the given limits set, we propose a mixed model of text localization in an image that involves preprocessing, detection, and post-processing. For detection, we choose a shortened convolutional neural network (CNN) type, which satisfies the specified restrictions. However, with a reduced SSD, the recognition quality may deteriorate. To support high-quality recognition, we offer a preprocessing step. At the preprocessing stage, the input RGB image is processed to enhance and expand the space of its features: remove unnecessary information (noise), change the contrast, brightness, color palette and other image processing procedures. At this stage, we can use structural methods (MSER, SURF, gradient method, Delaunay triangulation, Voronoi diagram, data structure usage, and NN) and in particular SWT. However, considering the time constraints, the classical SWT spends too much time searching for gestures of the same width. Therefore, optimization is required. In this work, we offer such SWT optimization using the Voronoi diagram, which allows you to accelerate and improve the quality of work significantly (to determine the contours of the gestures of their width), as well as to expand the space of features for further recognition.

The SSD is the most suitable CNN for our conditions. We have modified the SSD architecture (optimization of convolutional depth, size of the input image and feature map, number of channels of the input image). This allowed us to satisfy the specified limits (5Mb, 1-2 seconds). We conducted several tests with an SSD detector based on Inception v3 extractor features with a reduced number of neurons in each layer. Detection accuracy dropped by only 5%, while the number of operations 10 times decreased. In addition, the analysis showed that the most promising model for solving the problem of localization of a text on the image of mobile devices is a combined model that we choose as a basis. In order to get the maximum quality of localization of text at minimal expenses of computing resources, we propose to act according to the following scheme, Fig. 2.

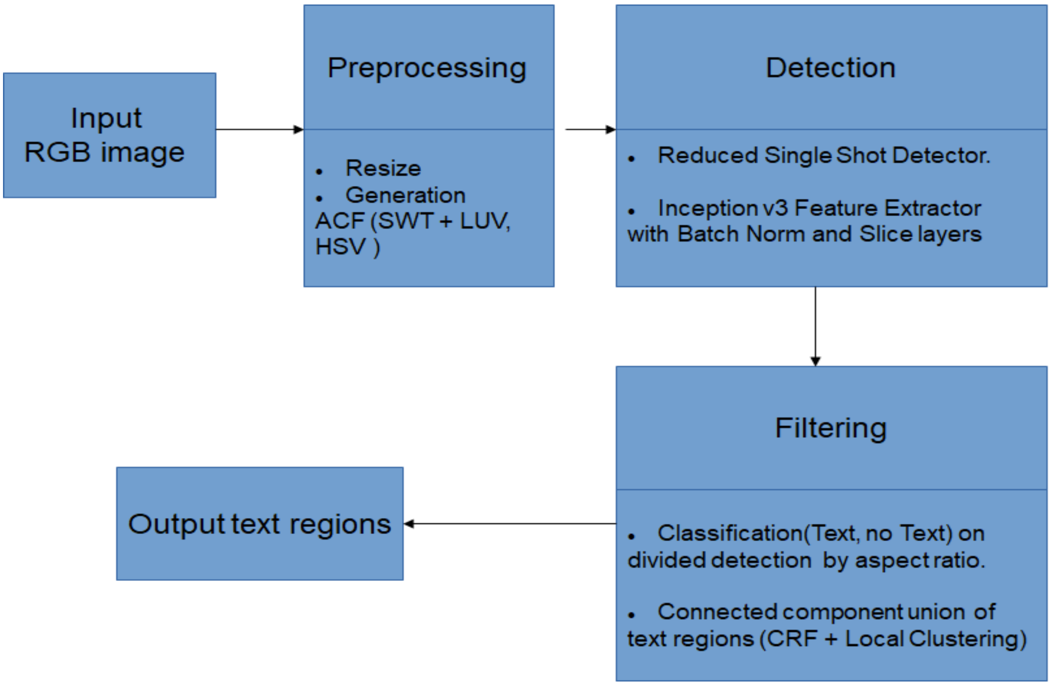

Input
RGB image
Preprocessing
• Resize
• Generation ACF (SWT + LUV, HSV)
Detection
• Reduced Single Shot Detector.
• Inception v3 Feature Extractor with Batch Norm and Slice layers
Filtering
• Classification(Text, no Text) on divided detection by aspect ratio.
• Connected component union of text regions (CRF + Local Clustering)
Output text regions

Figure 2: General scheme of algorithm for text detection on a natural image

Thus, we choose the following sequence of steps for the text localization algorithm on the natural image:

1) Expand the range of features. An expanded feature in our case is SWT. Finding a SWT depends on the quality of the algorithm for detecting borders, the key element of which is the threshold. To do this, we find the SWT at different threshold values and leave the maximum stable strokes (the maximum number of thresholds for which the strokes do not change their significance. Fig. 3).

2) Then, we submit to the input of the modified NN SSD RGB image and generated SWT. We submit SWT not only to the first input layer, but also to other layers for features at different scales. Fig. 4.

3) 3) At the post-processing stage, the text blocks detected are fed to the input of the filtering block. At this stage, the blocks that turned out to be false, as well as the separated parts of the words are combined into words and sentences.

4. Description of the Problem Solving Results

Already at the first stage of the development of the algorithm of localization (development of a prototype without optimization) we obtained optimistic results.

- Localization time on core i7 CPU - 0,4-0,7s; on ARM - 1-1.5s.
- Memory -1.9 MB and Accuracy – 0.61.
- NN training was taking place on a new Dataset that holds 17,000 photos (1-2-day training time).

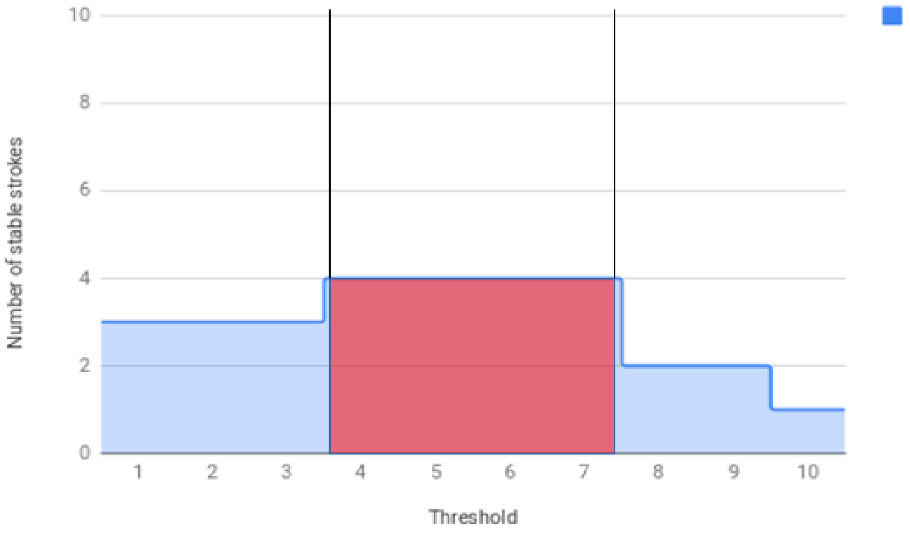


Figure 3: A stable stroke graphs. The red shows the zone of maximum stable

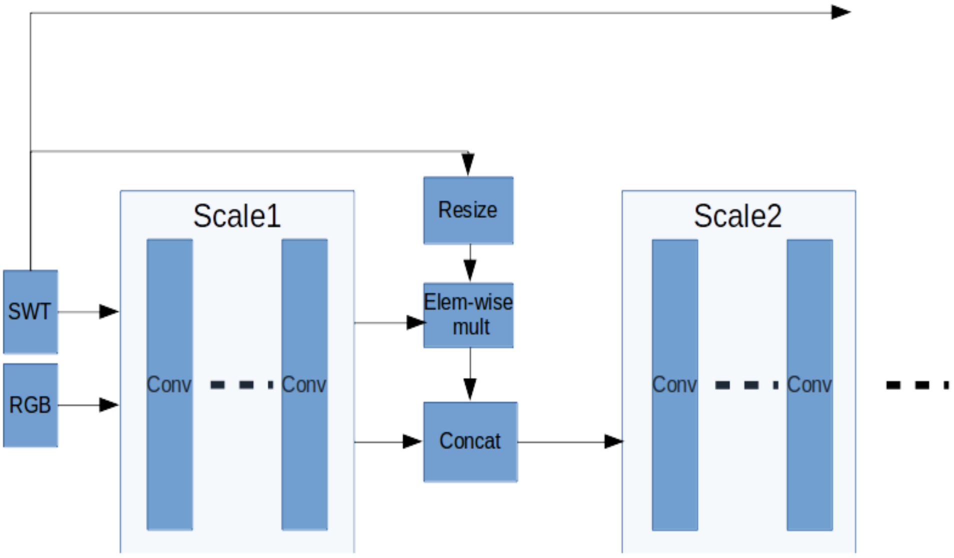


Figure 4: Using SWT to expand the NN features

4.1. SWT Optimization Method

To improve the quality and speed of recognition at the preprocessing stage, we proposed the optimization of the SWT algorithm, based on the Voronoi diagram and other procedures (Fig. 5).

In particular, to optimize SWT, we first apply one of the methods of binarization (for example, the Canny method [15]), which allows finding outlines based on their hierarchy (Fig. 6, a).

We remove extra points from the resulting contours (Fig. 6, b). To the resulting set of points that form contours, we use the Voronoi diagram.

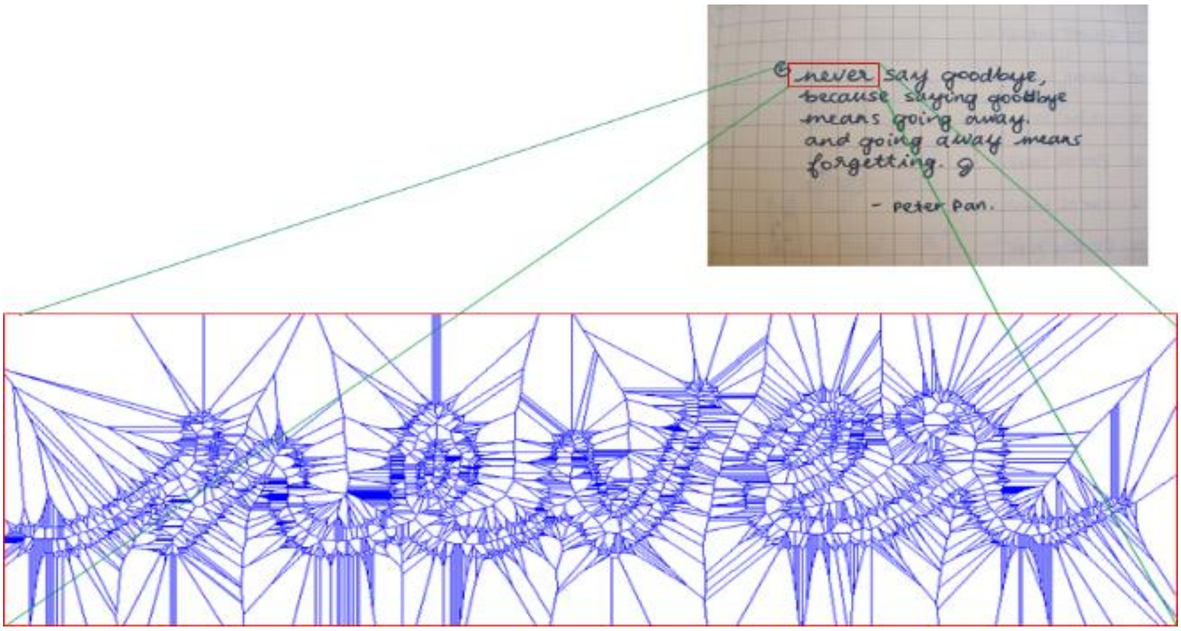
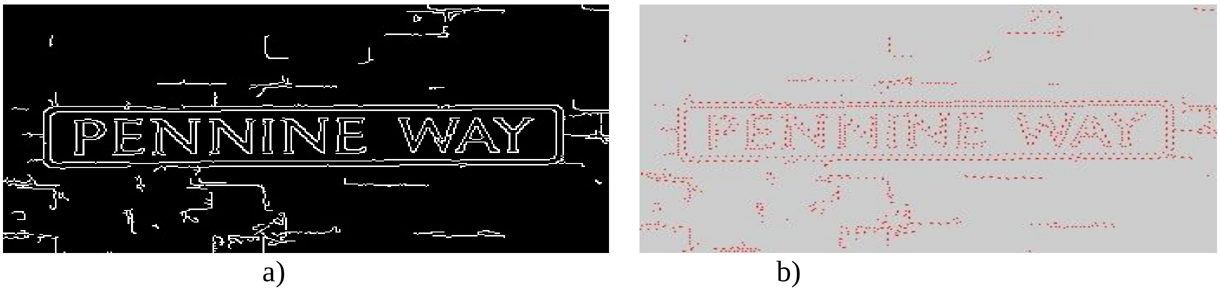


Figure 5: Optimized SWT based on segmentation using the Voronoi diagram


a)

b)

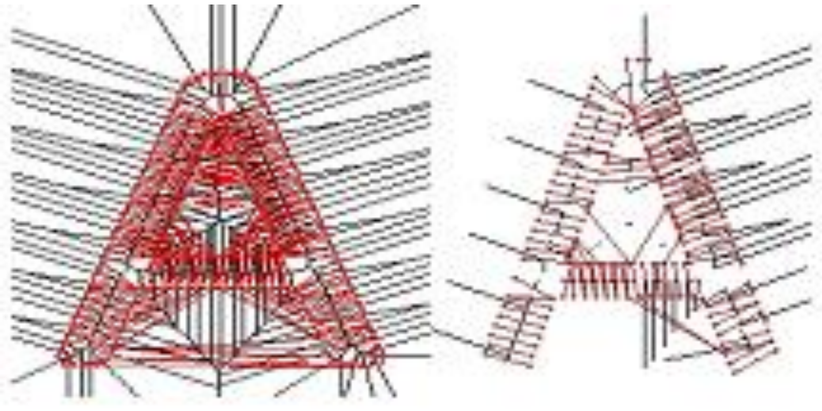
Figure 6: a) The result of the binarization algorithm; b) Contours taking into account the hierarchy and extraction of extra points

For this purpose, we can use one of the known effective algorithms, for example, the Fortune's algorithm [52] with the complexity of O (nlogn). In the process of constructing the Voronoi diagram, we present it as an expanded double–connected-edge-list (DCEL) [28], which contains distance between a pair of points for each edge that divides it. This allows you to define contours of the equal width (letters). If we have a Voronoi diagram for the set of input points in the form of an RSP, after binarization and deletion of noises, it is possible to localize the gestures of the letters in equal thickness (or within a certain width of the contour), (Fig. 7, a). In this case, the search is carried out not in the whole Voronoi diagram, but only in the direction of equal distances, that is in the direction of letters skeleton, (Fig. 7, b). Compared to the usual SWT, which is based on the circumference of the pixels contour, we move along the edges that divide the points forming the contour of the letter. Due to it, the number of operations to identify the letter is reduced. By combining the received components of the letter, we get its silhouette.

In [3] it is assumed that the letters and words in the image, as a rule, have a constant stroke width. But the proposed SWT optimization is not tied rigidly to the contour width and it allows the algorithm to work within its averaged width. Therefore, according to the authors, it is prospectively to use the algorithm SWT to detect such objects. The stroke width can be used not only as one of the features in the classification of areas, but also as a feature when combining the areas in the words.

4.2. General Description of the Configured SSD

To solve the detection problem, we developed a prototype algorithm based on a modified convolutional neural network SSD, which in its original form occupied 40 MB. The neural network has been modified and abbreviated by removing extra layers and neurons. In the final form, the neural network occupies (1.9 mb). It has a smaller initial number of classes; some layers have been added to the inception channel to improve the recognition of large objects. Wherein, despite this reduction of NN, the quality of text recognition on images is retained. The inception block contains three layers for forming image attributes. These layers correspond to matrices 1x1, 3x3 and 5x5. . To ensure the rapid operation of the detection algorithm, we optimized the Inception blocks. Such optimization involves adding Batch Normalization and Slice layers, as well as reducing the number of neurons based on statistics that determine the feasibility of using a neuron for a particular layer. The operating time of the MF - 0.4-0.7 s (it was on Everest i7000, CPU 0, 2 -0.8 s). A training base was also created on the basis of labeled test images, new added photos from the phone, as well as the Coco Dataset. Total amount of the training base of 17 000 pictures. The ability to submit more than 3 channels to HMs is provided through the Aggregate Channel Feature (ACF) [29].



a) b)

Figure 7: a) Localization of letters gestures in equal width; b) Search in the direction of skeleton letters

4.3. Comparative Characteristics of the SSD

We conducted a series of experiments to optimize the number of inception blocks and their effect on recognition. For this purpose, the statistics of localized bounding boxes were collected. Each bounding box was assigned to one of the clusters, which is responsible for a certain size of the detector. Also, compared to the original NN, the reduced network has a batch norm and slice layers. Table 1 shows the number of neurons for the original and reduced NN for each layer.

Table 1
Comparison between original NN and reduced

Layer name	Original NN	Reduced NN
conv1	64	13
conv2	192	38
inception_3a/output	256	45
inception_3b/output	480	83
inception_4a/output	512	91
inception_4b/output	512	-
inception_4c/output	512	90
inception_4d/output	528	142
inception_4e/output	832	142
inception_5a/output	832	142
inception_5b/output	1024	180
inception_6a/output	512	98
inception_7a/output	256	-

4.4. Post-processing (Filtering)

To solve this problem a prototype algorithm was developed on the basis of a small neural network in order to recognize the words of the same style (Fig. 8). This neural network is used at the stage of searching for words that have not been recognized by the main SSD neural network and that match the style with the selected high scored word.

We highlighted a large number of negatives in order to improve stylesheet recognition. Below are a few images (Fig. 9) on which the labeled version of the image is shown on the left and the result of the neural network operation is on the right, in addition the found text is classified as handwritten - red text blocks and printed - green.

In case of SWT modification is worth noting that we have developed a modified algorithm for constructing Voronoi diagram based on the idea of Fortune. The algorithm was adapted specifically for the SWT modification. Figure 10 shows graphs of the comparison of the algorithms of constructing the Voronoi diagram on the N data obtained after image binarization.

The blue color shows the execution time of the developed algorithm, and the orange color is the time to run the library algorithm with OpenCV. The graphs show that the developed algorithm is much faster than the library algorithm on large sets of input data.

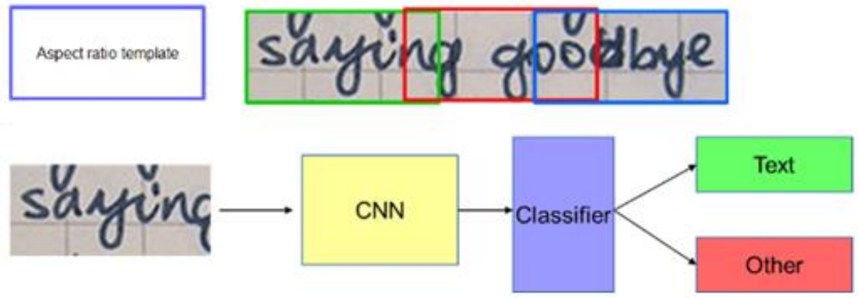


Figure 8: Example of style classification pipeline

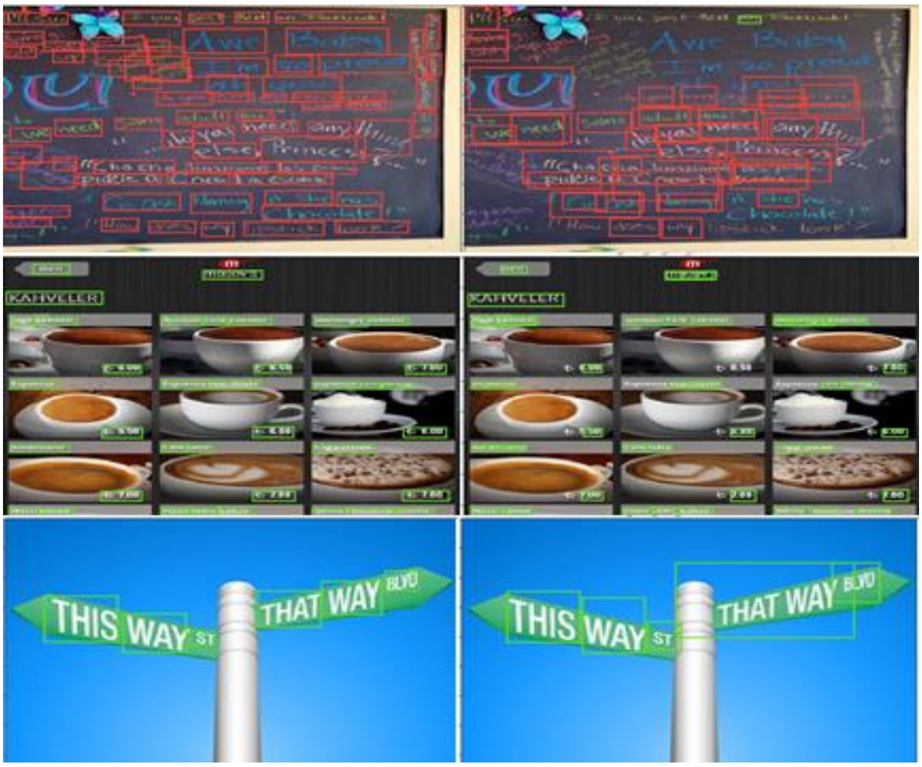


Figure 9: Examples of handwritten and printed text detection

4.5. Experimental Data

The comparison of proposed approaches was conducted on COCO-Text dataset (Tab. 2.). Accuracy of the proposed model by classes, Table 3:

Table 2
Comparison of different detection methods

Method	Recall	Precision	F-Score
Mod-SSD	0.287	0.455	0.351
Mod-SSD + SWT	0.331	0.498	0.397
Mod-SSD + SWT + Filtering	**0.365**	**0.521**	**0.429**

Table 3
Detection accuracy by image classes

Class	Accuracy
Letter	0.56
Street signs	0.72
Blackboard	0.67
Board	0.6
Card	0.61
Menu	0.56
Signboards	0.63

For each number of points the diagram was constructed 10 times, after which the average time for this number of points was determined. Below is a comparative characteristic for the average time of the algorithm, Tab. 4.

Table 4
Average speed improvement between SWT and Voronoi Diagram SWT

	AVG SWT (ms)	AVG VOR (ms)	Diff (ms)
KNU(4k resolution)	4166	2348	1818 (43%)
blackboard	1567	418	1149 (73%)
letter	385	272	112 (29%)
street_sign	633	221	412 (65%)
board	147	97	50 (34%)
signboards	129	84	45 (35%)
menu	142	119	23 (16%)
card	149	139	10 (6%)
TOTAL	1526	837	688

5. Conclusions

1. A new hybrid text detection approach for mobile devices is proposed, which is a combination of structural methods for the allocation of attributes and their application for the training of SSD. To ensure the rapid operation of the detection algorithm, the Inception blocks were optimized, which included adding Batch Normalization and Slice layers, as well as reducing the number of neurons based on statistical data that determine the feasibility of using a neuron for a particular layer.
2. A modified SWT method, which uses the Voronoi diagram to find the width of text symbols (gestures), is proposed. This approach considerably speeds up the work of SWT, because after filtering the ribs of the Voronoi diagram we work with a much smaller amount of data.

3. To combine text blocks into groups (lines), the small style classification NN is used, that allows to take into account not only the connection directly with the adjoining text block, but also with its environment. To generate the graph, for the first time, the Delaunay triangulation was used for centroids of text blocks.

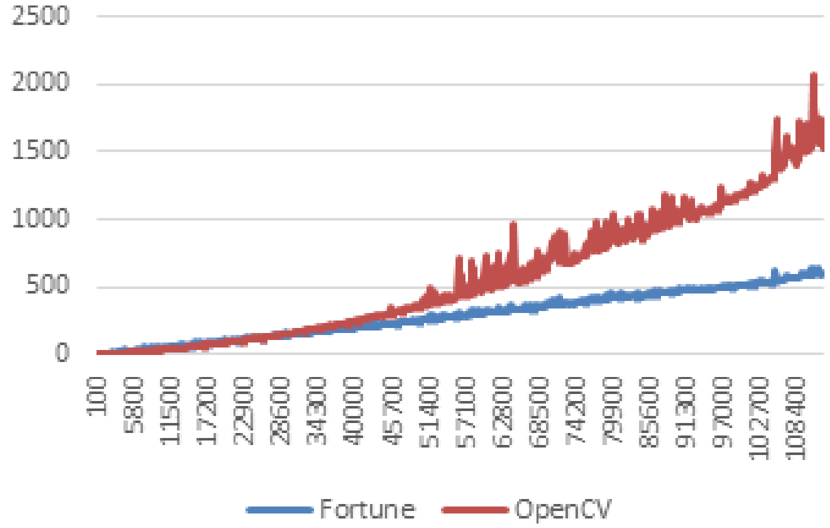


Figure 10: Comparison of the algorithms of constructing the Voronoi diagram on N data

6. References

[1] R. Gao, F. Shafait, S. Uchida, Y. Feng (2014). A Hierarchical Visual Saliency Model for Character Detection in Natural Scenes. LNCS, 8357, 18-29.

[2] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. Wu, A. Ng (2011). Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. In Proceedings of 11th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Beijing, China , pp. 440 – 445.

[3] B. Epshtein, E. Ofek, Y. Wexler (2010). Detecting Text in Natural Scenes with Stroke Width Transform. In Proceedings of 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, vol. 5, pp.2963-2970.

[4] Y. Kunishige, F. Yaokai, S. Uchida (2011). Scenery Character Detection with Environmental Context [Text]. In Proceedings of 11th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Beijing, China, pp. 1049 – 1053.

[5] S. Uchida, Y. Shigeyoshi, Y. Kunishige, F. Yaokai. Keypoint–Based Approach Toward Scenery Character Detection. In Proceedings of 11th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Beijing, China , pp. 819 –823, (2011).

[6] Y. Du, H. Ai, S. Lao (2011). Dot Text Detection Based on FAST Points. In Proceedings of 11th International Conference on Document Analysis and Recognition (ICDAR) IEEE, Beijing, China, pp. 435 – 439.

[7] C. Jung, Q.F. Liu, J. Kim (2009). Accurate text localization in images based on SVM output scores [Text]. Image and Vision Computing, 27, 1295–1301.

[8] Y. Deng, B.S. Manjunath, H. Shin (1999). Color Image Segmentation. In Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition , IEEE, Fort Collins, USA , V.2, pp. 446–451.

S. A. Hojjatoleslami and J. Kittler (1998). Region Growing: A New Approach. IEEE Trans. On Image Processing 7(7), 1079-1084. http://cgm.computergraphics.ru/content/view/147

[9] G. Koepfler, C. Lopez, J.M. Morel (1994). A Multiscale Algorithm for Image Segmentation by Variational Method SIAM. Journal on Numerical Analysis 31(1), 282–299.

[10] J.L. Marroquin. 1985. Probabilistic Solution of Inverse Problems Tech. Rep. Massachusetts Institute of Technology.

[11] L. G. Prewitt. 1970. Object Enhancements and Extraction. Picture Processing and Psychopictorics 10, 15-19. B. Lipkin and A. Rosenfeld (eds.), Academic Press, New York .

[12] R. M. Haralick, L. G. Shapiro (1985). Image Segmentation Techniques. Computer Vision, Graphics, and Image Processing, 29(1), 100-132.

[13] J. Canny (1986). A Computational Approach to Edge Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 8(6), 679-698.

[14] W. Huang, Y. Qiao, X. Tang (2014). Robust scene text detection with convolution neural network induced MSER trees. LNCS, 8692, 497-511.

[15] M. Delakis, Cr. Garcia (2008). Text detection with convolutional neural networks. In Proceedings of International Conference on Computer Vision Theory and Applications, January, Funchal, Madeira – Portugal, pp. 290-294.

[16] Y. Du, H. Ai, S. Lao (2011). Dot Text Detection Based on FAST Points. In Proceedings of International Conference on Document Analysis and Recognition, IEEE, Beijing, China, pp.435-440.

[17] B. H. Shekar, M. L. Smitha, S. Palaiahnakote (2014). Discrete Wavelet Transform and Gradient Difference Based Approach for Text Localization in Videos. In Proceedings of 5th International Conference on Signal and Image Processing, ICSIP, IEEE, Bangalore, India, South Korea, pp. 280-284.

[18] C. Enachescu, Cr. D. Miron (2009). Handwritten Digits Recognition Using Neural Computing. Scientific Bulletin of the Petru Maior University of Tirgu Mures 6 (XXIII), 17-20.

[19] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, J. Malik (2012). Semantic segmentation using regions and parts. In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, Providence, USA, pp. 3378 – 3385.

[20] T. Nguyen, S. Antoshchuk, A. Nikolenko, V. Sotov (2016). Correlation-extreme method for text area localization on images. In Proceedings of First International Conference on Data Stream Mining & Processing (DSMP), IEEE, Lviv, Ukraine , pp.173-176.

[21] W. Huang, Z. Lin, J. Yang, J. Wang (2013). Text localization in natural images using Stroke Feature Transform and Text Covariance Descriptors. In Proceedings of International Conference on Computer Vision (ICCV), IEEE, December, Sydney, Australia , pp. 1241-1248.

[22] X.- C.Yin, X. Yin, K. Huang, H.–W. Hao (2014). Robust Text Detection in Natural Scene Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(5), 970 – 983.

[23] L. Neumann, J. Matas (2016). Real-time Lexicon-free Scene Text Localization and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(9), 1872 – 1885.

[24] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C.L (2015). Tan Text Flow: A Unified Text Detection System in Natural Scene Images. In Proceedings of International Conference on Computer Vision (ICCV), December, Convention Center in Santiago, Chile, pp. 4651-4659.

[25] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, B. Girod (2011). Robust Text Detection in Natural Images with Edge-enhanced Maximally Stable Extremal Regions. In Proceedings of 18th IEEE International Conference on Image Processing, IEEE, Brussels, Belgium, pp. 2601 – 2604.

[26] F. Preparata, M.I. Shamos. 1985. Computational Geometry: An introduction. Springer-Verlag, Berlin.

[27] B. Yang, J. Yan, Z. Lei, S.Z. Li (2014). Aggregate channel features for multi-view face detection. In Proceedings of International Joint Conference on Biometrics, IEEE, Clearwater, USA, pp. 1-8.

[28] S. Fortune (1987). A sweep line algorithm for Voronoi diagrams. Algorithmica 2, 153-174.

[29] N.C. Kha and N. Masaki (2016). Enhanced Character Segmentation for Format-Free Japanese Text Recognition. In Proceedings of 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, P.R. China , pp. 138 – 143.

[30] K. C. Nguyen, M. Nakagawa (2016). Text-Line and Character Segmentation for Offline Recognition of Handwritten Japanese Text. IEICE technical report, pp.53-58.

[31] M. H. Nguyen, S-H. Kim, H. Yang, J., G.,S. Lee (2014). Stroke Width Based Skeletonization for Text Images. Journal of Computing Science and Engineering 8(3), 149-156.