# DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents

Rafael Dueire Lins
UFRPE/UFPE, Recife, Brazil
rdl@cin.ufpe.br

Steven J. Simske
Colorado State University, USA
Steve.Simske@colostate.edu

Rodrigo Barros Bernardino
UFPE, Recife, Brazil
rbb4@cin.ufpe.br

## ABSTRACT

Document image binarization is a key process in many document processing platforms. The DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents assessed the performance of eight new algorithms and also 41 other "classical" algorithms. Besides the quality of the binary image, the execution time of the algorithms was assessed. The evaluation dataset is composed of 32 documents photographed using four widely-used mobile devices with the strobe flash *on* and *off*, under several different angles of capture.

## KEYWORDS

Binarization, Quality evaluation, Performance evaluation, Documents

**ACM Reference Format:**
Rafael Dueire Lins, Steven J. Simske, and Rodrigo Barros Bernardino. 2020. DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents. In *ACM Symposium on Document Engineering 2020 (DocEng '20), September 29-October 2, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3395027.3419578

## 1 INTRODUCTION

The conversion of a color image into its black-and-white version is called *binarization* or *thresholding*. Such a process is of paramount importance in the pipeline of many document processing systems. No binarization algorithm is good for all kinds of text document images [11]. This competition focuses on the binarization of camera-acquired text documents, the kind of document that is most often photographed. The assessment performed using 32 text documents, photographed under three slightly different angles, with four different models of hand-held cell phones, with their in-built strobe flash *on* and *off*. Two quality measures were used: the proportion between the black-to-white pixels in the photographed to the scanned binary version of the documents and an OCR-based transcription. Besides the quality of the produced image, the binarization time is of great importance to determine the applicability of a binarization algorithm, thus this feature is also assessed here.


*DocEng '20, September 29-October 2, 2020, Virtual Event, CA, USA*

ACM ISBN 978-1-4503-8000-3/20/09.
https://doi.org/10.1145/3395027.3419578

## 2 PARTICIPANTS

Six teams enrolled in this competition, one of which submitted three different binarization algorithms. They are listed in enrollment order with the affiliation is of their first member:

**(A) University of Groningen, The Netherlands** *(Sheng He and Lambert Schomaker)*: This program is based on Tensorflow and the algorithm DeepOtsu [7]. The neural network is trained to learn the degradations in document images and to iteratively produce uniform images. The binarization map is obtained through a global Otsu threshold.

**(B) Hubei University of Technology, China** *(Wei Xiong, Dichun Yang, Meihui Ai, Ling Yue, Lei Zhou, Min Li, Song Wang):* The rationale behind this method is: (1) a downsampling operation decreases the spatial resolution of the feature map; (2) the upsampling operation attempts to restore the feature map leading to a texture smoothing. The dilated convolutional layer allows all the intermediate feature maps to have exactly the same spatial resolution as its input. The proposed hybrid dilation rate setting can maintain or even increase each convolutional layer receptive field size, thus effectively improving the segmentation accuracy.

**(C) Sugarcube Information Technology Sàrl, Switzerland** *Jean-Luc Bloechle*: The Yin Yang binarization algorithm detects the background of the original image using small overlapping windows. First, each window calculates its median color using a quantized color palette. Then, the estimated background image is generated by interpolating the computed median pixels of the overlapping windows. Next, the background image is subtracted from the original image, and the resulting difference image is transformed into grayscale, keeping only the lowest RGB component. A binarization is done by Otsu's algorithm. Detection and removal of small isolated connected components is made.

**(D) West Pomeranian University of Technology, Poland** *(Hubert Michalak, Krzysztof Okarma)*: An approximation of illumination distribution of the background is made by images downsampling, yielding a loss of details related to shapes of individual characters. After resizing the downsampled image to the original resolution using the same kernel, the image containing only the low-frequency information is obtained, representing the approximated high-resolution background. This image is subtracted from the original image, enhancing the text data, followed by a simple increase in contrast and logical negation.

**(E) University of Alicante, Spain** *(Jorge Calvo-Zaragoza and Antonio Javier Gallego)*: Image binarization is treated as a two-class classification task at the pixel level. [4]. A Convolutional Neural Networks is trained to label an input image pixel-by-pixel, taking into account its neighbors. Several pixels can be processed at the same time, leading to higher efficiency. A thresholding process converts the scores into binary values.

**(F) Universitas Syiah Kuala, Indonesia** *(Khairun Saddami)*:

**Method iNICK:** Extends the NICK binarization method [20]. The image standard deviation is used to determine the $k$ value as $k = -\sigma/(255 - 1.5\sigma)$, where $\sigma$ is the image standard deviation that represents the image contrast.

**Method CNW:** Combination of Niblack and Wolf [18]. The threshold $T = (2m + mk((\sigma/m) - (\sigma/R) - 1))/2$, where $\sigma$ is the image standard deviation, $m$ is the mean of local window, $R$ is the maximum standard deviation, $k = 0.35$.

**Method CLD** Combined the local adaptive and global thresholding formulas, as described in [19].

**(G) Traditional Algorithms** Forty-one previously published binarization algorithms have also been considered in the analysis. Twenty-three of them are among the top ten in the different measures of quality analysis: Bataineh [1], Bernsen [2], Bradley [3], DocDLink [28], DocUNet [12], Ergina-G [23], Gattal [5], Huang [13], ISauvola [6], IsoData [26], JiaShi [8], Li-Tam [10], Minimum[17], Moments [25], Nick [9], Otsu [16], Prewitt [17], Sauvola [21], Singh [22], Su-Lu [24], WAN [15], Wolf [27], Lu-Su [14].

## 3 TEST SET

The binarization of photographed documents is far more difficult than scanned ones as the photo resolution varies between devices and is non-uniform due to differences in the distance between the document and the camera. It may also suffer interference from external light sources and even non-uniform illumination from the in-built strobe flash. This test set encompasses 32 documents that are part of the DIB dataset (https://dib.cin.ufpe.br), obtained from four different models of portable cell-phones (Motorola Moto Z2, Apple Iphone 6, Apple Iphone SE, and Samsung Galaxy N4), whose specifications are presented in Table 1. Besides the device model, the documents in this set were clustered according to having the in-built strobe-flash set as "*on*" or "*off*". Figure 1 presents samples of the documents used in this test set.

**Table 1: Summary of device camera specifications**

| | Moto Z2 | Iphone 6 | Iphone SE | Galaxy N4 |
|---|---|---|---|---|
| Megapixels | 12 | 8 | 12 | 16 |
| Flash | Dual led | Dual led | Dual led | Dual led |
| Aperture size | f/1.7 | f/2.2 | f/2.2 | f/2.2 |
| Sensor size | - | 1/3 inch | 1/3 inch | 1/2.6 inch |
| Pixel size | 1.4 $\mu$m | 1.22 $\mu$m | 1.22 $\mu$m | 1.12 $\mu$m |

## 4 QUALITY EVALUATION METHODS

Two quality measures were used here to evaluate the performance of the binarization algorithms. The first one compares the proportion between the black-to-white pixels in the scanned and photographed binary documents [11] and is here denoted by $P_{err}$ . The

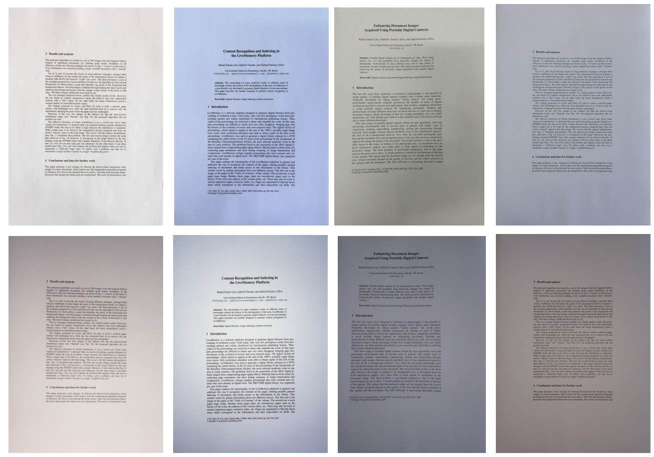

**Figure 1: Samples of the images clustered by device (iPhone 6, iPhone SE, Moto Z2, Samsung Galaxy N4) and set-up of the strobe flash (top-line "off", bottom-line "on").**

second one made use of the Tesseract 4.1.1 OCR to transcribe the documents and applies the Levenshtein distance ($L_{dist}$) to the correct number of characters in the document transcription ($\#char$ ). Thus, the error rate is calculated as: $[L_{dist}] = (\#char - L_{dist})/\#char$. The measures were ranked in the same way as in [12]. First, the ranking for each measure is calculated for each document in a class. Then, the summation of the rank order for all documents in the class defines the final ranking. Visual inspection was applied to check the consistency of the results obtained.

## 5 PROCESSING-TIME EVALUATION

The purpose of the processing time evaluation here is to provide an order of magnitude of time elapsed for binarizing the whole datasets. The training-times for the AI-based algorithms were not computed. The processing device was:

- CPU: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
- RAM: 32GB
- GPU: GeForce GTE 1650 4GB

The competing algorithms were implemented using two operating systems and different programming languages, for specific hardware platforms such as GPUs.

- **Windows 10 (version 1909):**
  - DeepOtsu (team A): Python 3.6 with GPU
  - YinYangFilter (team C): Java 14
  - HMKO (team D): Matlab 2018a
  - team E: Python 3.6 with GPU
  - team F: Matlab 2017a
- **Linux Pop!_OS 20.04:**
  - Dilated-UNet (team B): Python 3.6 with GPU

Some experiments have been done with the algorithms that can be executed on both OS, and no significant processing time difference has been noticed. That is likely due to the fact that the exact same hardware and up to date compilers have been used in all cases. Although the user programming languages were Matlab, Python and Java, it is known that they are often used as an API to lower level implementations, leading to smaller differences in time purely due to the programming language used. Also, modern compilers are all nearly equally efficient. That can be easily verified by the results, as some algorithms using GPU performed fast, while others were slow. As for the Matlab implemented ones, some were among the fastest, while others among the slowest methods. Thus, if some optimization is made, using modern versions of the compilers, most algorithms should have similar performance in different languages.

The mean processing time was used in the analysis. The primary purpose is to provide the order of magnitude time of the processing time elapsed.

## 6 RESULTS

The DocEng'2020 Time-quality binarization competition assessed the quality of binary document images produced by forty-nine algorithms, eight new and forty-one "classical" ones. The evaluation was performed considering the quality of the produced images and the processing time.

The quality was measured based on the proportions of black-to-white pixels ($P_{err}$ ) in the digitally generated document and a new OCR-error count based on the Levenshtein distance ($[L_{dist}]$), which measures the similarity between the OCR produced text and the original text. The mean processing time was taken to evaluate the order of magnitude of the time complexity of the algorithms. For all measures, the mean value is presented and the ranking is decided at the individual image level.

Several conclusions may be drawn:

(1) The best ranked algorithms vary according to the conditions of the captured images, reinforcing the claim that no binarization algorithm is good for all document images.
(2) In most cases, the classical algorithms were dominant among the best ranked; however, in several situations, as for Motorola Z2, one or more of the new algorithms presented similar quality results with the top three, but processed the images much faster.
(3) Using the same device and capture angle, but turning the strobe flash *on* or *off* strongly impacts the performance of the algorithms, leading to completely different rankings. The classical algorithms are the most impacted, as the new algorithms appearing in the flash ON rank still appear in the same device flash OFF rank. The classical algorithms often drop enough positions to leave the top from the ranking, either using $[L_{dist}]$ or $P_{err}$ as an error measure.

This competition provides another evidence that "no binarization algorithm is good for all types of document images".

## REFERENCES

[1] Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, and Khairuddin Omar. 2011. An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recognition Letters* 32, 14 (oct 2011), 1805–1813.
[2] J Bernsen. 1986. Dynamic thresholding of gray-level images. In *International Conference on Pattern Recognition*. 1251–1255.
[3] Derek Bradley and Gerhard Roth. 2007. Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools* 12, 2 (jan 2007), 13–21.
[4] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. 2019. A selectional auto-encoder approach for document image binarization. *Pattern Recognition* 86 (feb 2019), 37–47.
[5] Abdeljalil Gattal, Faycel Abbas, and Mohamed Ridda Laouar. 2018. Automatic Parameter Tuning of K-Means Algorithm for Document Binarization. In *Proceedings of the 7th International Conference on Software Engineering and New Technologies - ICSENT 2018*. ACM Press, New York, New York, USA, 1–4.
[6] Zineb Hadjadj, Abdelkrim Meziane, Yazid Cherfa, Mohamed Cheriet, and Insaf Setitra. 2016. ISauvola: Improved Sauvola's Algorithm for Document Image Binarization. Lecture Notes in Computer Science, Vol. 3212. Springer Berlin Heidelberg, Berlin, Heidelberg, 737–745.
[7] Sheng He and Lambert Schomaker. 2019. DeepOtsu: Document Enhancement and Binarization using Iterative Deep Learning. *Pattern Recognition* 91 (jan 2019), 379–390.
[8] Fuxi Jia, Cunzhao Shi, Kun He, Chunheng Wang, and Baihua Xiao. 2018. Degraded document image binarization using structural symmetry of strokes. *Pattern Recognition* 74 (feb 2018), 225–240.
[9] Khurram Khurshid, Imran Siddiqi, Claudie Faure, and Nicole Vincent. 2009. Comparison of Niblack inspired binarization methods for ancient documents. In *SPIE Proceedings*, Kathrin Berkner and Laurence Likforman-Sulem (Eds.). 72470U.
[10] C.H. Li and P.K.S. Tam. 1998. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters* 19, 8 (1998), 771–776.
[11] Rafael Dueire Lins, Rodrigo Barros Bernardino, Darlisson Marinho de Jesus, and Jose Mario Oliveira. 2017. Binarizing Document Images Acquired with Portable Cameras. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 45–50.
[12] Rafael Dueire Lins, Ergina Kavallieratou, Elisa Barney Smith, Rodrigo Barros Bernardino, and Darlisson Marinho de Jesus. 2019. ICDAR 2019 Time-Quality Binarization Competition. In *2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 1539–1546.
[13] Di Lu, Xin Huang, and LiXue Xue Sui. 2018. Binarization of degraded document images based on contrast enhancement. *International Journal on Document Analysis and Recognition* 21, 1-2 (jun 2018), 123–135.
[14] Shijian Lu, Bolan Su, and Chew Lim Tan. 2010. Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition (IJDAR)* 13, 4 (dec 2010), 303–314.
[15] Wan Azani Mustafa and Mohamed Mydin M. Abdul Kader. 2018. Binarization of Document Image Using Optimum Threshold Modification. *Journal of Physics: Conference Series* 1019, 1 (jun 2018), 012022.
[16] Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66.
[17] Judith M. S. Prewitt and Mortimer L. Mendelsohn. 2006. THE ANALYSIS OF CELL IMAGES. *Annals of the New York Academy of Sciences* 128, 3 (dec 2006), 1035–1053.
[18] Khairun Saddami, Putri Afrah, Viska Mutiawani, and Fitri Arnia. 2018. A New Adaptive Thresholding Technique for Binarizing Ancient Document. In *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*. IEEE, 57–61.
[19] Khairun Saddami, Khairul Munadi, Yuwaldi Away, and Fitri Arnia. 2019. Effective and fast binarization method for combined degradation on ancient documents. *Heliyon* (2019).
[20] Khairun Saddami, Khairul Munadi, Sayed Muchallil, and Fitri Arnia. 2017. Improved Thresholding Method for Enhancing Jawi Binarization Performance. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 1108–1113.
[21] J. Sauvola, M. Pietikäinen, and M Pietikainem. 2000. Adaptive document image binarization. *Pattern Recognition* 33, 2 (2000), 225–236.
[22] T. Romen Singh, Sudipta Roy, O. Imocha Singh, Tejmani Sinam, and Kh. Manglem Singh. 2011. A New Local Adaptive Thresholding Technique in Binarization. *IJCSI International Journal of Computer Science Issues* 08, 6 (dec 2011), 271–277.
[23] Vavilis Sokratis, Ergina Kavallieratou, Roberto Paredes, and Kostas Sotiropoulos. 2011. A Hybrid Binarization Technique for Document Images. In *Studies in Computational Intelligence*. 165–179.
[24] Bolan Su, Shijian Lu, and Chew Lim Tan. 2010. Binarization of historical document images using the local maximum and minimum. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems - DAS '10*. ACM Press, New York, New York, USA, 159–166.
[25] Wen-Hsiang Tsai. 1985. Moment-preserving thresolding: A new approach. *Computer Vision, Graphics, and Image Processing* 29, 3 (1985), 377–393.
[26] Flavio R. Velasco. 1979. *Thresholding Using the Isodata Clustering Algorithm*. Technical Report. OSD or Non-Service DoD Agency. 14 pages.
[27] Christian Wolf, Jean-Michel Jolion, and Françoise Chassaing. 2003. Text localization, enhancement and binarization in multimedia documents. In *Object recognition supported by user interaction for service robots*, Vol. 2. IEEE Comput. Soc, 1037–1040.
[28] Lichen Zhou, Chuang Zhang, and Ming Wu. 2018. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.

**Table 2: Final Results**

| | Pixel Proportion | | | | | | OCR Error | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OFF | | | ON | | | OFF | | | ON | | |
| # | Alg. | $P_{err}$ | Time (s) | Alg. | $P_{err}$ | Time (s) | Alg. | $L_{dist}$ | Time (s) | Alg. | $L_{dist}$ | Time (s) |
| **Apple iPhone 6** | | | | | | | | | | | | |
| 1 | D | 0.54 | 0.10 | Bradley | 0.67 | 0.22 | IsoData | 0.98 | 0.03 | WAN | 0.97 | 0.70 |
| 2 | F1 | 0.52 | 3.45 | F1 | 0.57 | 3.15 | Otsu | 0.98 | 0.01 | Bataineh | 0.97 | 0.08 |
| 3 | Minimum | 0.52 | 0.03 | A | 0.77 | 2,529.66 | Li-Tam | 0.97 | 0.03 | D | 0.97 | 0.04 |
| 4 | Bradley | 0.64 | 0.22 | ISauvola | 0.84 | 0.26 | Huang | 0.97 | 0.04 | F1 | 0.97 | 3.15 |
| 5 | Singh | 0.55 | 0.26 | D | 0.64 | 0.04 | D | 0.97 | 0.10 | Nick | 0.97 | 0.11 |
| 6 | C | 0.64 | 1.08 | Bernsen | 0.77 | 1.73 | F1 | 0.98 | 3.45 | Bradley | 0.97 | 0.22 |
| 7 | Wolf | 0.61 | 0.15 | Su-Lu | 0.72 | 1.07 | Wolf | 0.98 | 0.15 | ISauvola | 0.97 | 0.26 |
| 8 | Nick | 0.58 | 0.12 | C | 0.72 | 1.17 | Minimum | 0.97 | 0.03 | F2 | 0.97 | 3.44 |
| 9 | IsoData | 0.66 | 0.03 | WAN | 0.86 | 0.70 | Bradley | 0.97 | 0.22 | Singh | 0.97 | 0.24 |
| 10 | Gattal | 0.65 | 76.19 | F3 | 0.77 | 4.14 | Moments | 0.97 | 0.03 | JiaShi | 0.97 | 14.23 |
| 11 | ISauvola | 0.87 | 0.28 | Lu-Su | 0.90 | 98.22 | Prewitt | 0.97 | 0.03 | Su-Lu | 0.97 | 1.07 |
| **Apple iPhone SE** | | | | | | | | | | | | |
| 1 | F1 | 0.51 | 3.20 | F1 | 0.63 | 3.42 | IsoData | 0.98 | 0.03 | Singh | 0.98 | 0.27 |
| 2 | C | 0.61 | 1.29 | Bradley | 0.76 | 0.25 | Otsu | 0.98 | 0.01 | Nick | 0.98 | 0.12 |
| 3 | D | 0.66 | 0.04 | D | 0.74 | 0.03 | Li-Tam | 0.97 | 0.03 | ISauvola | 0.98 | 0.28 |
| 4 | Minimum | 0.53 | 0.08 | Nick | 0.68 | 0.12 | Huang | 0.97 | 0.04 | Bradley | 0.98 | 0.25 |
| 5 | Nick | 0.50 | 0.12 | Singh | 0.69 | 0.27 | D | 0.97 | 0.10 | D | 0.98 | 0.03 |
| 6 | Singh | 0.50 | 0.28 | Bernsen | 0.77 | 1.93 | F1 | 0.98 | 3.45 | F1 | 0.98 | 3.42 |
| 7 | Wolf | 0.61 | 0.18 | C | 0.76 | 1.42 | Wolf | 0.98 | 0.15 | JiaShi | 0.98 | 15.89 |
| 8 | Bradley | 0.75 | 0.25 | ISauvola | 0.84 | 0.28 | Minimum | 0.97 | 0.03 | Wolf | 0.98 | 0.17 |
| 9 | Bernsen | 0.68 | 1.94 | Su-Lu | 0.71 | 1.19 | Bradley | 0.97 | 0.22 | Bataineh | 0.97 | 0.09 |
| 10 | Sauvola | 0.55 | 0.13 | A | 0.85 | 2,723.68 | Moments | 0.97 | 0.03 | F2 | 0.98 | 3.46 |
| 11 | Gattal | 0.60 | 74.68 | Bataineh | 0.90 | 0.09 | Prewitt | 0.97 | 0.03 | WAN | 0.97 | 0.78 |
| **Motorola Moto Z2** | | | | | | | | | | | | |
| 1 | Bernsen | 0.60 | 2.52 | F1 | 0.56 | 3.28 | D | 0.98 | 0.05 | F2 | 0.98 | 3.60 |
| 2 | D | 0.64 | 0.05 | Wolf | 0.60 | 0.21 | IsoData | 0.98 | 0.16 | Bradley | 0.98 | 0.31 |
| 3 | ISauvola | 0.69 | 0.38 | Bernsen | 0.65 | 2.31 | Wolf | 0.98 | 0.22 | Bataineh | 0.98 | 0.11 |
| 4 | Bradley | 0.64 | 0.32 | D | 0.64 | 0.04 | Singh | 0.97 | 0.37 | IsoData | 0.97 | 0.13 |
| 5 | Wolf | 0.59 | 0.22 | Bradley | 0.66 | 0.31 | Nick | 0.97 | 0.16 | Nick | 0.97 | 0.15 |
| 6 | F1 | 0.58 | 3.33 | F3 | 0.72 | 4.52 | Sauvola | 0.97 | 0.16 | ISauvola | 0.98 | 0.35 |
| 7 | A | 0.79 | 3,986.92 | ISauvola | 0.74 | 0.35 | DocDLink | 0.97 | 215.13 | Prewitt | 0.97 | 0.13 |
| 8 | E | 0.77 | 13.60 | Singh | 0.67 | 0.36 | Otsu | 0.97 | 0.02 | Singh | 0.97 | 0.36 |
| 9 | DocDLink | 0.79 | 215.13 | C | 0.70 | 1.37 | Prewitt | 0.97 | 0.16 | D | 0.98 | 0.04 |
| 10 | DocUNet | 0.79 | 131.13 | DocDLink | 0.84 | 205.36 | A | 0.97 | 3,986.92 | Otsu | 0.97 | 0.01 |
| 11 | Singh | 0.69 | 0.37 | Lu-Su | 0.83 | 125.01 | Bradley | 0.97 | 0.32 | Sauvola | 0.97 | 0.15 |
| **Samsung Galaxy N4** | | | | | | | | | | | | |
| 1 | ISauvola | 0.58 | 0.40 | Bradley | 0.98 | 0.34 | E | 0.58 | 14.46 | Bataineh | 0.98 | 0.12 |
| 2 | D | 0.61 | 0.05 | ISauvola | 0.98 | 0.40 | ISauvola | 0.56 | 0.40 | WAN | 0.98 | 1.11 |
| 3 | Bradley | 0.61 | 0.34 | Bataineh | 0.98 | 0.12 | DocUNet | 0.60 | 133.84 | F1 | 0.98 | 3.43 |
| 4 | DocUNet | 0.65 | 131.95 | D | 0.98 | 0.05 | F3 | 0.59 | 5.00 | Bradley | 0.98 | 0.35 |
| 5 | E | 0.67 | 14.23 | IsoData | 0.98 | 0.17 | F1 | 0.55 | 3.43 | F2 | 0.98 | 3.48 |
| 6 | F1 | 0.57 | 3.35 | F1 | 0.98 | 3.35 | Bradley | 0.65 | 0.35 | Wolf | 0.98 | 0.23 |
| 7 | F3 | 0.64 | 4.86 | Otsu | 0.98 | 0.02 | Ergina-G | 0.64 | 52.33 | D | 0.98 | 0.05 |
| 8 | Lu-Su | 0.65 | 176.43 | F2 | 0.98 | 3.42 | D | 0.64 | 0.05 | Otsu | 0.97 | 0.02 |
| 9 | Ergina-G | 0.67 | 51.31 | Nick | 0.98 | 0.17 | DocDLink | 0.63 | 229.31 | IsoData | 0.97 | 0.17 |
| 10 | Moments | 0.70 | 0.17 | Moments | 0.98 | 0.17 | Lu-Su | 0.69 | 167.78 | ISauvola | 0.97 | 0.40 |
| 11 | Bernsen | 0.76 | 2.63 | WAN | 0.98 | 1.10 | Ergina-L | 0.64 | 52.56 | JiaShi | 0.97 | 22.92 |