An OCR-based Application using Tesseract Engine to Extract Text Information from Ultrasound B-MODE Images

Jiří Blahuta[1,*,†], Tomas Soukup[1,†], Jiří Kozel[2,†] and Jiří Martinů[1,†]

[1]Institute of Computer Science, Faculty of Philosophy and Science in Opava, Bezručovo náměstí 1150/13, 74601 Opava, Czech Republic
[2]Faculty of Medicine, University of Ostrava, Ostrava, Syllabova 19, 70300 Ostrava-Vítkovice, Czech Republic

Abstract
In this paper we introduce our developed OCR-based application focused on the extraction of text data from ultrasound B-MODE images. These images contain not only visual image but also additional, important text information about the examination, image data, etc. The extraction of these data is helpful in clinical practice. The application has a simple, user-friendly interface to use. The core of the algorithm to extract the text data is focused on Tesseract engine with C# programming language; front-end is a Windows Forms application. Although this software is fast, simple and user-friendly, in some cases, the recognition could produce a error, like patient's name incorrectly recognized and/or some characters are missing or mistaken. That means, some text in the image could be missed or misinterpreted in comparison with the original. However, while the software could be a helpful tool, the recognition accuracy should be improved. After that, the application can be used for different types of medical images, e.g. CT/CTA, PET, SPECT and many more. Currently achieved accuracy is about 90 % in average. The authors discuss some ideas to increase the accuracy of the recognition and also some front-end features that can be improved for more comfortable use. Extracted text data can be saved as a CSV file for further processing.

Keywords
image text extraction, medical OCR application, C# Tesseract engine, ultrasound B-MODE OCR

1. A brief introduction to OCR

The first generation of *Optical Character Recognition* (OCR) were developed in 90s in parallel with increased using computer graphics (including computer vision) and in general, digital data processing. In general, we can define the OCR principle as a technique how to extract non-editable information to an editable form from a digital document; the information can be stored as an editable form to further processing. The most typical task for OCR is an extraction of text from a bitmap file, e.g. photos, scans, etc. In other words, the algorithms of OCR systems are based on recognition of each character and its conversion into a letter. One of the most difficult tasks is the recognition of handwritten letters due to their uniqueness for each people. Nowadays, especially for non-handwriting text, the OCR is relatively simple due to existing large databases of known commonly used fonts, etc.

A few years ago, many OCR systems were developed using artificial neural network models for learning and training the samples, like Error Back-Propagation models, recurrent neural networks, convolutional neural networks, etc. A lot of artificial neural networks have been developed for OCR tasks to improve efficacy and speed of the processing to achieve more and more correct results. Currently and also in future, there is a new progressive way for OCR; artificial intelligence (AI). The AI in parallel with deep learning methods is rapidly progressive in many fields of image processing and computer vision. AI approach can be used to create a lot of non-supervised learning neural network models, especially convolutional networks, to more effective, faster and more accurate processing. Recently, AI approach and deep learning are one of the most progressive fields in many tasks like image processing, speech recognition, generated realistic graphics from keywords, ChatGPT for writing almost perfect articles from predefined keywords (prompts) and many more. OCR is currently a perspective part for AI approach.

More information about OCR algorithms and techniques in image processing you can find for example in a guide [1].

Software tools for OCR can be as a stand-alone applications, web applications online and also a module (plug-in) in many complex software. There are a lot of web online tools with variable output quality. Our goal is to design a simple software tool for OCR focused on radiology.

ITAT'24: Information technologies – Applications and Theory, September 20–24, 2024, Drienica, Slovakia
*Corresponding author.
†These authors contributed equally.
jiri.blahuta@fpf.slu.cz (J. Blahuta); tsoukup@centrum.cz (T. Soukup); jiri.kozel@upol.cz (J. Kozel); jiri.martinu@fpf.slu.cz (J. Martinů)
https://www.slu.cz/fpf/en/ (J. Blahuta); https://www.slu.cz/fpf/en/ (T. Soukup); https://lf.osu.eu (J. Kozel); https://www.slu.cz/fpf/en/ (J. Martinů)

2. Radiological images and text content inside

Our study is focused on OCR for radiological images. Simply, we would like to have a tool to extract text non-editable information inside the image. An example of the image containing embedded, non-editable text, is shown in Fig. 1. Text information, which should be extracted, is marked with a yellow border.

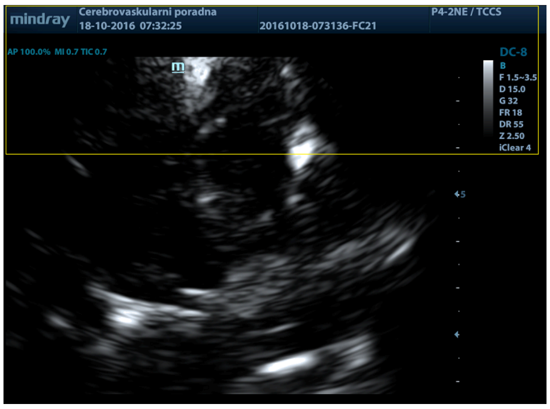


Figure 1: Embedded text in an ultrasound digital image

Indeed, the goal is to extract the text inside the border (Fig. 1) to an editable form. These information can consist very important values about the patient, image settings, etc.

In radiology worldwide, we have DICOM standard for storing, sharing and viewing radiological images which also contains DICOM metadata that can be extracted in many software tools for DICOM imaging purposes. Although, sometimes we need to have an option to extract text data directly from the image without using metadata and/or we have only a bitmap image (like screens or exported from DICOM original) in which no DICOM metadata are not stored. Thanks to this fact, we need to have our tool to direct reading and storing the text data from the image.

3. Developed application to extract embedded text

Our developed application is a simple, lightweight tool for extraction the text information, like in Fig. 1, into the editable form stored into a CSV file. GUI of the application is simple, see Fig. 2.

There are only four buttons:

- to select an input image
- to extract the text using OCR

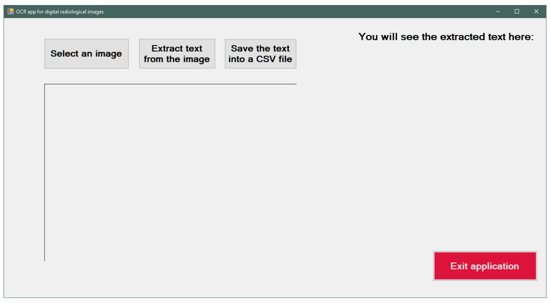

OCR app for digital radiological images
Select an image
Extract text from the image
Save the text into a CSV file
You will see the extracted text here:
Exit application

Figure 2: GUI of our developed application after running

- to save extracted text into a CSV
- to close the application

and one PictureBox and one TextBox. That means, using the application is really user-friendly.

The application has been developed in C# programming language and compiled into an EXE file in Microsoft Visual Studio IDE. It perfectly demonstrates how to use Tesseract library for OCR. There are no specific requirements for running, you must have Windows 7 of higher and installed .NET Framework 4.0 or higher (which is normally pre-installed in all recent OS Windows versions). So you can run the application without installing anything.

3.1. The background of used technologies

The application has been developed as a Windows Forms application in C#. There are two inseparable parts; programming of core algorithm for OCR and visual design of GUI. For example, this is the button to open system dialog window for loading a new image with a condition if an error appears:

```
private void selectimg(object sender,
    EventArgs e)
{
    try
        {
        OpenFileDialog file = new
            OpenFileDialog();
        file.Filter = @"Bitmap|*.jpg;*.
            jpeg;*.png;*.bmp|DICOM|*.dcm|
            All|*.*";
        file.Title = "Select an image for
             the text extraction";
            if (file.ShowDialog() ==
                System.Windows.Forms.
                DialogResult.OK)
                {
                pictureBox1.Image = new
                    Bitmap(file.FileName)
                    ;
                }
            }
        catch (Exception ex)
            {
            MessageBox.Show(@"Error loading
                image." + ex.Message);
            }
}
```

From this code segment, the dialog to open a file, deals BMP, PNG and JPEG bitmap formats image format and another formats can be added to file.Filter line. However, there is a bug with DICOM format. Commonly the images used in clinical practice are in DICOM format natively. Conversion into a bitmap format may decrease the image quality. The input image can be in color or in grayscale.

3.1.1. Core OCR-based algorithm with Tesseract

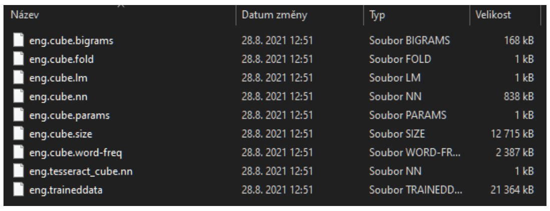
The core algorithm used in this application is based on OCR image processing using Tesseract library which is one of the most powerful OCR libraries. The engine is based on unsupervised learning used a convolutional neural network (CNN) principle. In general, CNN are a progressive way how to use deep learning algorithms in many fields in which unsupervised learning is applicable like recognition, prediction, etc. More details about CNN theoretical background you can find for example here [2].

To use it, we must declare the namespace by using Tesseract directive; after install the library. The button to extract the text from loaded image has the following function:

```
private void extract(object sender,
    EventArgs e)
{
    try
        {
         if (pictureBox1.Image != null)
         {
         Bitmap img = new Bitmap(
             pictureBox1.Image);
         TesseractEngine engine = new
             TesseractEngine("./tessdata"
             , "eng", EngineMode.Default)
             ;
         Page page = engine.Process(img,
             PageSegMode.Auto);
         textBox1.Text = page.GetText();
         }
        else
         {
         MessageBox.Show(@"No image
             selected.");
         }
        }
    catch (Exception ex)
        {
            MessageBox.Show("Error converting
                image." + ex.Message);
            }
}
```

We can see the lines to execute Tesseract engine for OCR processing from the path where all necessary files are stored (tessdata folder). In these files, all neural network configuration to recognition each character, are defined.

We use the tessadata package which is one of three available models; tessdata-best (best accuracy should be achieved but slow) and tessadata-fast (fast but may be inaccurate). The tessdata package contains LSTM and legacy. Detailed description of the packages is available at Tesseract Github website [3].



Název	Datum změny	Typ	Velikost
eng.cube.bigrams	28.8. 2021 12:51	Soubor BIGRAMS	168 kB
eng.cube.fold	28.8. 2021 12:51	Soubor FOLD	1 kB
eng.cube.lm	28.8. 2021 12:51	Soubor LM	1 kB
eng.cube.nn	28.8. 2021 12:51	Soubor NN	838 kB
eng.cube.params	28.8. 2021 12:51	Soubor PARAMS	1 kB
eng.cube.size	28.8. 2021 12:51	Soubor SIZE	12 715 kB
eng.cube.word-freq	28.8. 2021 12:51	Soubor WORD-FR...	2 387 kB
eng.tesseract_cube.nn	28.8. 2021 12:51	Soubor NN	1 kB
eng.traineddata	28.8. 2021 12:51	Soubor TRAINEDD...	21 364 kB

Figure 3: Tesseract engine training data for English language

The first condition to execute the OCR, the Picture Box is not empty (some image must be loaded). Eventually we can add a whitelist of characters which should be recognized like this line for digits only:

```
ocr.SetVariable("tessedit_char_whitelist"
    , "0123456789");
```

We currently used English definition files (so "eng" in the line of the code segment above). The majority of the information n US images are in English or abbreviations, but for example, patient names are in Czech language. That means, we can also download Czech settings from Github repositories [4].

In the application, Tesseract ver. 4.1 is used. It is an open-source OCR engine that can be used in many programming languages like C++, C#, Java, Python, etc and in addition, it is a multi-platform solution. In 2005 HP released Tesseract as an open-source software developed by Google since 2006. You can find comprehensive information about Tesseract in C# at website [5].

If the text is extracted successfully, displayed in a textbox on the right side, see Fig. 4.

3.1.2. Saving Extracted Text into a CSV File

After the text extraction, there is function to save the text into a CSV file using System.IO namespace. See the code segment with try-catch exception:

```
private void save(object sender,
    EventArgs e)
{
    try
        {
            if (textBox1.Text != "You
                will see the extracted
                text here:")
                {
                string path = "extract.
                    csv";
                string output = textBox1.
                    Text;
                File.WriteAllText(path,
                    output.ToString());
                textBox2.Text = "Text
                    saved into extract.
                    csv file.";
                }
            else
                {
                MessageBox.Show("No text
                    to save.");
                }
            }
    catch (Exception ex)
        {
        MessageBox.Show("Error saving
            text into a file." + ex.
            Message);
        }
}
```

In this code segment, there is defined filename and the message in the textbox about successful saving the file. If there is no extracted text available, an error message appears.

3.2. Known limitations and ideas to improve

This is the first version of the application but there are two known limitations. Basic ideas to improve the application:

- the CSV file has a predefined name extract.csv so when we save a new one, the file is overwritten without any warning
- when we try to open a some DCM (DICOM image) file, the error occurs (invalid image.Parameter)

Th first limitation regarding CSV output file should be fixed by string.path changing. There is a fixed filename.

Impossibility of DICOM loading is given by properties of the PictureBox element; it is possible to view only a bitmap file. We want to find a different way how to load DICOM files to OCR processing. Currently we are working on solution based on conversion from DICOM to bitmap before loading the image into PictureBox element. There are to options:

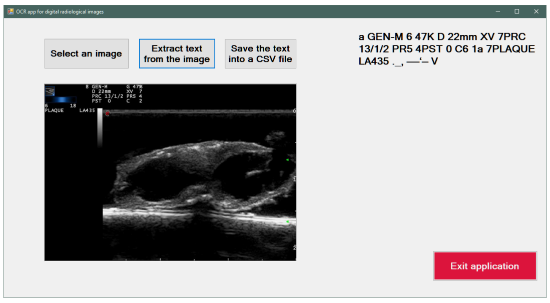
- automatic conversion into a bitmap format when DICOM is selected in the open dialog
- to add an extra button to convert DICOM file into a bitmap with the same name and save it

The second option should be more simple. So, we will get the bitmap file to load into PictureBox to processing. For conversion there are some libraries in C# like RasterEdge, Aspose.Imaging or EvilDICOM.

The recognition accuracy is discussed below.

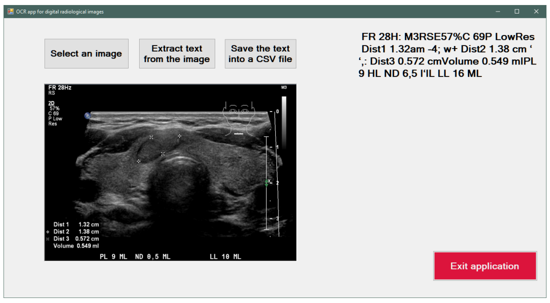
4. Achieved accuracy level of the recognition to judge reliability

In this part we will discuss achieved results. Although the application is lightweight and easy to use, the results are usable but not perfect. The following two examples; loaded image and the extracted text in Fig. 4 and in Fig 5.


OCR app for digital radiological images
Select an image
Extract text from the image
Save the text into a CSV file
Exit application

```
a GEN-M 6 47K D 22mm XV 7PRC
13/1/2 PR5 4PST 0 C6 1a 7PLAQUE
LA435 ._, —'– V
```

Figure 4: The first example - US B-MODE image of an atherosclerotic plaque in-vitro


OCR app for digital radiological images
Select an image
Extract text from the image
Save the text into a CSV file
Exit application

```
FR 28H: M3RSE57%C 69P LowRes
Dist1 1.32am -4; w+ Dist2 1.38 cm '
',: Dist3 0.572 cmVolume 0.549 mlPL
9 HL ND 6,5 I'IL LL 16 ML
```

Figure 5: The second example - US B-MODE image of the thyroid gland

In both cases, there are some errors. Some parts of the text are missing and/or are incorrectly interpreted, e.g. "B GEN-M" vs. "a GEN-M" or missing "z" in 28 Hz.

Table 1
Errors of OCR results in four different image types loaded

image type	text with no errors	errors found
UZ B-MODE TCS	no	missing/mistaken characters/numbers, spaces
UZ B-MODE TCS	no	missing/mistaken characters/numbers, spaces
UZ B-MODE TCS	no	missing/mistaken characters/numbers, patient's name, spaces
UZ B-MODE TCS	yes	-
UZ B-MODE thyroid	no	missing/mistaken characters/numbers, patient's name, spaces
UZ B-MODE thyroid	no	missing/mistaken characters/numbers, image labels, spaces
UZ B-MODE thyroid	no	missing/mistaken characters/numbers, spaces
UZ B-MODE thyroid	no	missing/mistaken characters/numbers, image labels, spaces
UZ B-MODE TCS type 2	no	missing/mistaken characters/numbers, no date examination, spaces
UZ B-MODE TCS type 2	no	missing/mistaken characters/numbers, no date examination
UZ B-MODE TCS type 2	no	missing/mistaken characters/numbers, patient's name, spaces
UZ B-MODE TCS type 2	no	missing/mistaken characters/numbers, spaces
UZ B-MODE TCS type 2	no	missing/mistaken characters/numbers, no date examination, spaces
UZ B-MODE ATS	no	missing/mistaken characters/numbers, patient's name, spaces
UZ B-MODE ATS	yes	-
UZ B-MODE ATS	yes	-
UZ B-MODE ATS	yes	-
UZ B-MODE ATS	no	missing/mistaken characters/numbers, spaces

Authors have some sets of US transcranial images and also US images of thyroid gland and neighbor structures; these images were used in many author's previous studies focused on evaluation of echogenicity level of substantia nigra for Parkinson's Disease [6, 7] and further studies on atherosclerotic plaques progress evaluation [8] using segmentation approach and the measurement of mediotemporal lobe atrophy to evaluate possibility of Alzheimer Disease progress [9, 10] based on black-white pixels ratio with computed threshold value. Finally, the last research has been focused on distinguishing between malignant and benign gland thyroid tumors [11]. All mentioned studies were performed from technical solution and also in clinical point of view.

So back to our OCR solution. To judge the reliability of the algorithm, 200 images have been randomly selected. That means we have compared the text in the image and extracted one to observe the quality of OCR algorithm. The following Tab. 1 summarizes the observed results.

In result, we would like to improve the accuracy of the text recognition. We can use different settings of the Tesseract or to find a different OCR engine. As we mentioned in the abstract, currently achieved accuracy is about 90 %, this is not perfect to use in clinical practice.

5. Conclusions and suggestions to improve the software

The presented application demonstrates how to use OCR engine Tesseract in a Windows Forms application developed in C# programming language. The goal is to create an application to OCR recognition of text information in radiological images from various modalities like diagnostic ultrasound, CT used in radiology. The application was developed in 2021 within a bachelor thesis. GUI of the application is user-friendly to use. This version has been developed for ultrasound B-MODE images but various images can be processed.

However, the achieved results show that OCR feature is applicable for radiological images but there are some errors with character recognition. In few cases, some characters were missing or incorrectly recognized. That means, it could have a error like incorrect patient name recognition. But the overall accuracy of the recognition is commonly exceeds 90 %. The best accuracy was achieved for UZ B-MODE TCS for which the application was tested firstly.

So, there are three key ideas to improve the application:

- to find a solution to better OCR accuracy
- to find a way how to import different types of DICOM modalities
- to find a solution for saving CSV file with any filename, not only predefined one

After that, the appication should be more comfortable and usable in clinical practice for different US images. In addition, the program can load not only ultrasound images, the same algorithm can be used OCR extraction from CT, MR and another image types. We would like to continue with developing.

References

[1] PARASHAR, K.A., KUMAR, E.A. *Feature Extraction Based Document Image Processing for OCR.* Lambert Academic Publishing, 68 p., 2017. ISBN: 978-6202062053.

[2] AGGARWAL, C.C. *Convolutional Neural Networks.* In: Neural Networks and Deep Learning. Springer, Cham, 2018, pp. 315 - 371.

[3] Trainedddata Files for Tessearact at Github, https://tesseract-ocr.github.io/tessdoc/Data-Files.html, last accessed 2024/07/05

[4] Github repositories for Tesseract in C#, https://github.com/tesseract-ocr/, last accessed 2024/07/05

[5] How to use Tesseract OCR in C#, https://dev.to/mhamzap10/how-to-use-tesseract-ocr-in-c-9gc, last accessed 2024/07/05

[6] BLAHUTA, J., BÁRTOVÁ, P., JELÍNKOVÁ, M., ČERMÁK, P., HERZIG, R., ŠKOLOUDÍK, D. *A new program for highly reproducible automatic evaluation of the substantia nigra from transcranial sonographic images.* Biomedical Papers, 2014, 158(4), pp. 621 - 627. DOI:10.5507/bp.2013.029.

[7] BLAHUTA, J., SOUKUP, T. MARTINŮ, J. *An Expert System Based on Using Artificial Neural Network and Region-Based Image Processing to Recognition Substantia Nigra and Atherosclerotic Plaques in B-Images: A Prospective Study.* In: Rojas I., Joya G., Catala A. (Eds.) Advances in Computational Intelligence. IWANN 2017. Lecture Notes in Computer Science, Vol. 10305, pp. 236 – 245. Springer, Cham. DOI: 10.1007/978-3-319-59153-7_21.

[8] BLAHUTA, J., SOUKUP, T., SOSÍK, P. *Approach to automatic segmentation of atherosclerotic plaque in B-images using active contour algorithm adapted by convolutional neural network to echogenicity index computation.* Paper presented at the CEUR Workshop Proceedings, 2020, Vol. 2718, pp. 223 - 229. ISSN: 1613-0073.

[9] BLAHUTA, J., SOUKUP, T. *The reproducibility assessment of black-white pixels ratio in transcranial B-images of medial temporal lobe in application for detectable feature of Alzheimer's disease.* In Proceedings of the International Conference on Biomedical Innovations and Applications, BIA 2020, pp. 53 - 57. DOI: 10.1109/BIA50171.2020.9244520.

[10] ŠKOLOUDÍK, D., KRULOVÁ, P., KISVETROVÁ, H., HERZIG, R., BLAHUTA, J., *Transcranial sonography of the medial temporal lobe in Alzheimer's disease patients.* [Transkraniální sonografie mediotemporálního laloku u pacientů s Alzheimerovou demencí]. Ceska a Slovenska Neurologie a Neurochirurgie, 2020, 83(2), 189-193. DOI:10.14735/amcsnn2020189.

[11] BLAHUTA, J., SOUKUP, T., LAVRINČÍK, J., PAVLÍK, L., REPASKÁ, Z. *Measurable Difference Between Malignant and Benign Tumor of the Thyroid Gland Recognizable Using Echogenicity Index in Ultrasound B-MODE Imaging: An Experimental Blind Study.* Bioinformatics and Biomedical Engineering: 9th International Work-Conference, IWBBIO 2022, pp. 283 - 296. ISBN: 978-3-031-07703-6, DOI: 10.1007/978-3-031-07704-3_23.

[12] POŠTULKA, R. *Rozpoznávání textu z medicínských obrazových dat.* Online. Bakalářská práce. Moravská vysoká škola Olomouc. 2021. Available at: https://is.mvso.cz/th/oeu8p/