

Tłumacz alfabetu Braille'a

Aplikacja umożliwiająca tłumaczenie tekstu (z obrazu) zapisanego angielskim alfabetem Braille'a na tradycyjny alfabet.

Programowanie w Java (PJAVA), semestr 20Z

autorzy: Zuzanna Adamiuk & Piotr Kielak

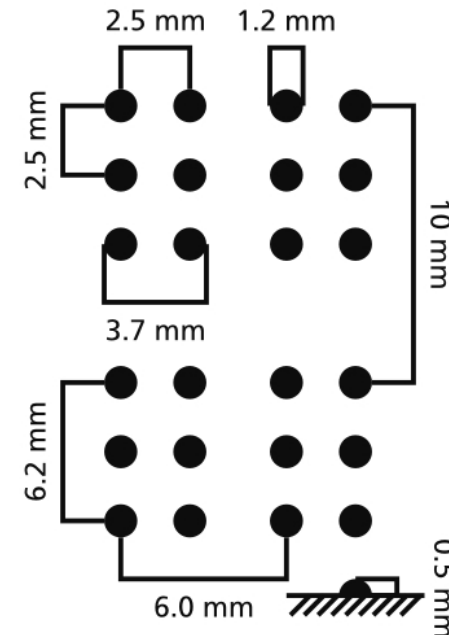
Założenia

Aplikacja umożliwia

- wczytywanie obrazów w formacie .jpg/.png z poziomu konsoli
- analizę znaków zapisanych alfabetem Braille'a (tłumaczenie, ilość znaków, średnie parametry itp.)
- zapisywanie wyników do pliku .txt

Wymagania

- w kadrze może znajdować się jedynie tekst
- tekst musi być równy względem krawędzi obrazu
- tekst musi spełniać normy przewidziane dla tekstu w alfabecie Braille'a (rys. obok)



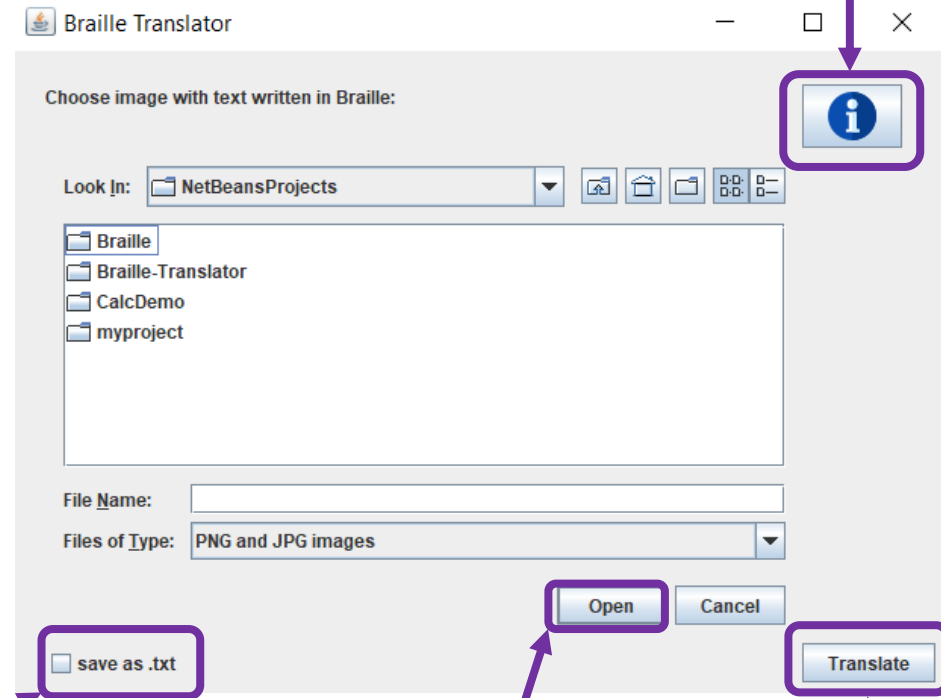
Schemat działania

- wczytanie obrazu przez użytkownika (poprzez interfejs graficzny)
- przetworzenie obrazu (część wykonywana jest automatycznie przez wykorzystaną metodę HoughCircles)
- tłumaczenie na podstawie danych z obrazu
- wyświetlenie i zapisanie przetłumaczonego tekstu w pliku .txt (użytkownik wybiera, czy chce jedynie przetłumaczyć tekst, czy również zapisać go do pliku)

Wczytywanie obrazu – interfejs graficzny

```
JFileChooser fc=new JFileChooser();
mainPanel.add(fc);
fc.setFileSelectionMode(JFileChooser.FILES_ONLY);
fc.setAcceptAllFileFilterUsed(false);
FileNameExtensionFilter filter = new FileNameExtensionFilter("PNG "
    + "and JPG images", "png", "jpg");
fc.addChoosableFileFilter(filter);

ActionListener actionChooseFile = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        File file = fc.getSelectedFile();
        path = file.getAbsolutePath();
        JOptionPane.showMessageDialog(null,
            "The image was successfully chosen!");
    }
};
fc.addActionListener(actionChooseFile);
```



Opis działania aplikacji

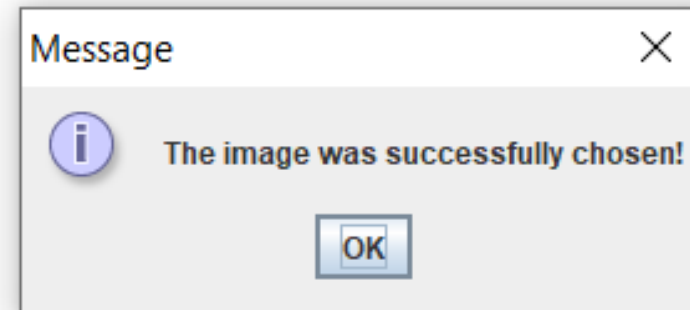
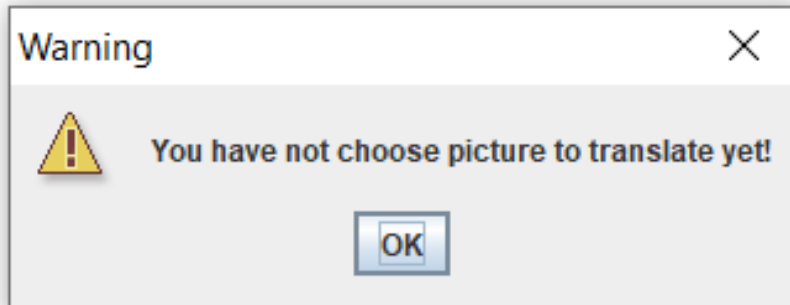
Możliwość wyboru, czy tłumaczenie ma być zapisane do pliku

Zatwierdzenie wybranego obrazu

Wyświetlanie tłumaczenia w nowym oknie

Komunikaty dla użytkownika

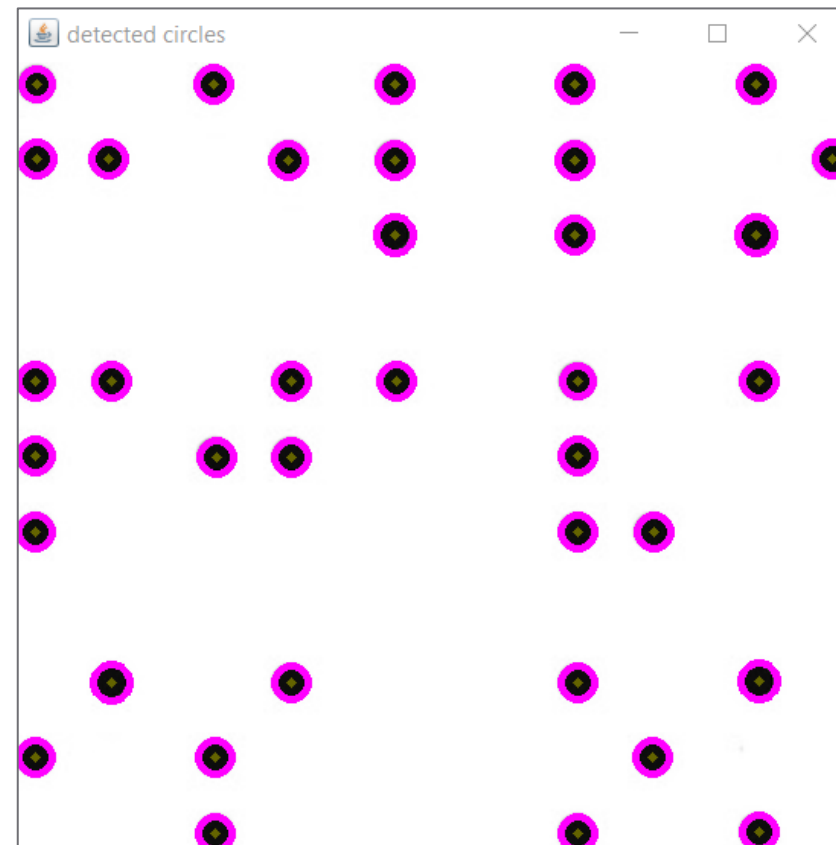
Aplikacja informuje użytkownika o obecnym stanie aplikacji
– informuje o błędach oraz o wykonanych operacjach.



Przetwarzanie obrazu

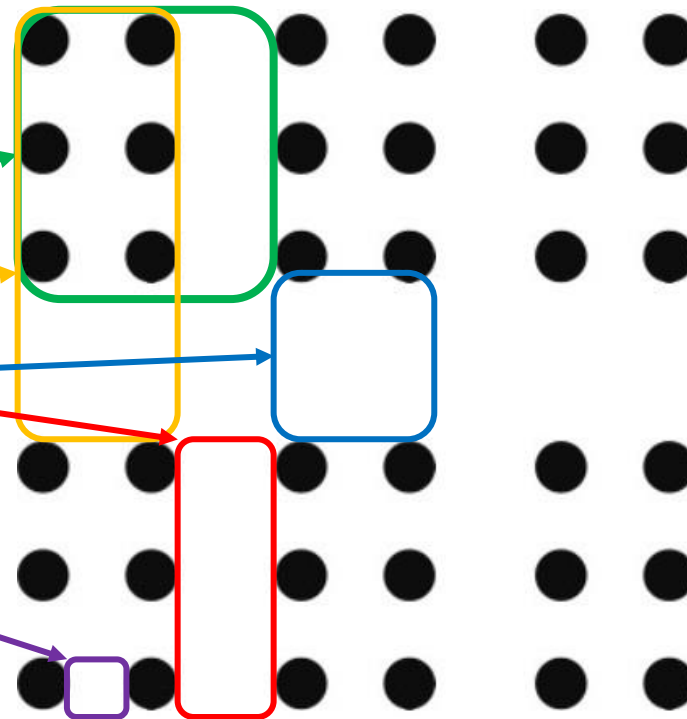
Wykorzystano metodę HoughCircles z biblioteki OpenCV 4.5.0.

```
// We are using HoughCircles method from OpenCV lib, which allows
// us to detect circles in edited image matrix
Mat circles = new Mat();
Imgproc.HoughCircles(imgEdited, circles, Imgproc.HOUGH_GRADIENT, 1.0,
    10, // value allowing to detect circles close to each other
    // (the lower it is, the closer circles can be detected)
    100.0, 30.0,
    1, // minimal radius to detect
    100); // maximal radius to detect
```



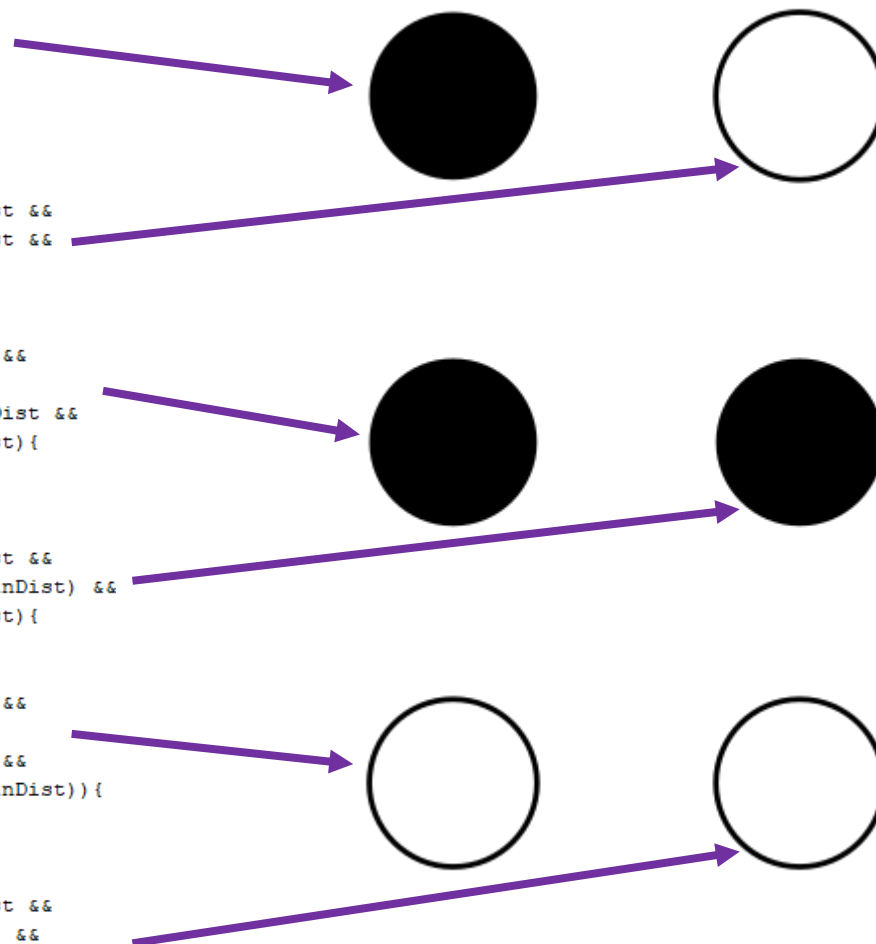
Algorytm tłumaczący

```
oneCharacterWithSpace = minDist * 6 / 2.5;  
oneCharacterWithNewLine = minDist * 10 / 2.5;  
lineSpace = minDist * 2.3 / 2.5;  
rowSpace = minDist * 2.5 / 2.5;  
dotSize = minDist * 1.2 / 2.5;  
dotSpace = minDist * 1.3 / 2.5;
```



Algorytm tłumaczący

```
// Algorithm translating a single character
if (pointArray[j].x < (currCol-1)*oneCharacterWithSpace + minDist &&
    pointArray[j].x > (currCol-1)*oneCharacterWithSpace &&
    pointArray[j].y < (currRow-1)*oneCharacterWithNewLine + minDist &&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine){
    currCharacter[0] = "1";
}
else if (pointArray[j].x < currCol*oneCharacterWithSpace &&
    pointArray[j].x > ((currCol-1)*oneCharacterWithSpace) + minDist &&
    pointArray[j].y < (currRow-1)*oneCharacterWithNewLine + minDist &&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine){
    currCharacter[1] = "1";
}
else if (pointArray[j].x < (currCol-1)*oneCharacterWithSpace + minDist &&
    pointArray[j].x > ((currCol-1)*oneCharacterWithSpace) &&
    pointArray[j].y < (currRow-1)*oneCharacterWithNewLine + 2*minDist &&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine + minDist){
    currCharacter[2] = "1";
}
else if (pointArray[j].x < currCol*oneCharacterWithSpace &&
    pointArray[j].x > ((currCol-1)*oneCharacterWithSpace) + minDist &&
    pointArray[j].y < (currRow-1)*oneCharacterWithNewLine + 2* (minDist) &&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine + minDist){
    currCharacter[3] = "1";
}
else if (pointArray[j].x < (currCol-1)*oneCharacterWithSpace + minDist &&
    pointArray[j].x > ((currCol-1)*oneCharacterWithSpace) &&
    pointArray[j].y < (currRow*oneCharacterWithNewLine - rowSpace)&&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine + 2* (minDist)){
    currCharacter[4] = "1";
}
else if (pointArray[j].x < currCol*oneCharacterWithSpace &&
    pointArray[j].x > ((currCol-1)*oneCharacterWithSpace) + minDist &&
    pointArray[j].y < (currRow*oneCharacterWithNewLine - rowSpace) &&
    pointArray[j].y > (currRow-1)*oneCharacterWithNewLine + 2* (minDist)){
    currCharacter[5] = "1";
}
```




Algorytm tłumaczący

Aktualnie tłumaczony znak jest przechowywany w tablicy currCharacter: [0,0,0,0,0,0], która zostaje zmieniona na ciąg znaków typu String – currCode: „000000”.

```
currCode = String.join("", currCharacter);

// Displaying current character code - just to check whether it's translated correctly
//System.out.println(currCode);

for (int p = 0; p < Letters.numofletters; p++){
    if (currCode.equals(Letters.idletters[p])){
        translated_text =translated_text + Letters.trueletters[p];
        break;
    }
}
```



```
public class Letters {
    public static String[] idletters = {"100000","101000","110000","110100","100100","111000","111100","101100","011000","011100",
        "100010","101010","110010","110110","100110","111010","111110","101110","011010","011110",
        "100011","101011","110011","110111","100111","111011","111111","101111","011011","011111",
        "100001","101001","110001","110101","100101","111001","111101","101101","011101",
        "001000","001010","001100","001101","001001","001110","001111","001011","000110","010010",
        "010110","010011","010111","000000","000010"};

    public static String[] trueletters = {"a","b","c","d","e","f","g","h","i","j",
        "k","l","m","n","o","p","q","r","s","t",
        "u","v","x","y","z","and","for","of","the","with",
        "ch","gh","sh","th","wh","ed","er","ou","w",
        "ea","bb","cc","dd","en","ff","gg","?", "in","st",
        "ar","ing","ble"," ",""," "};

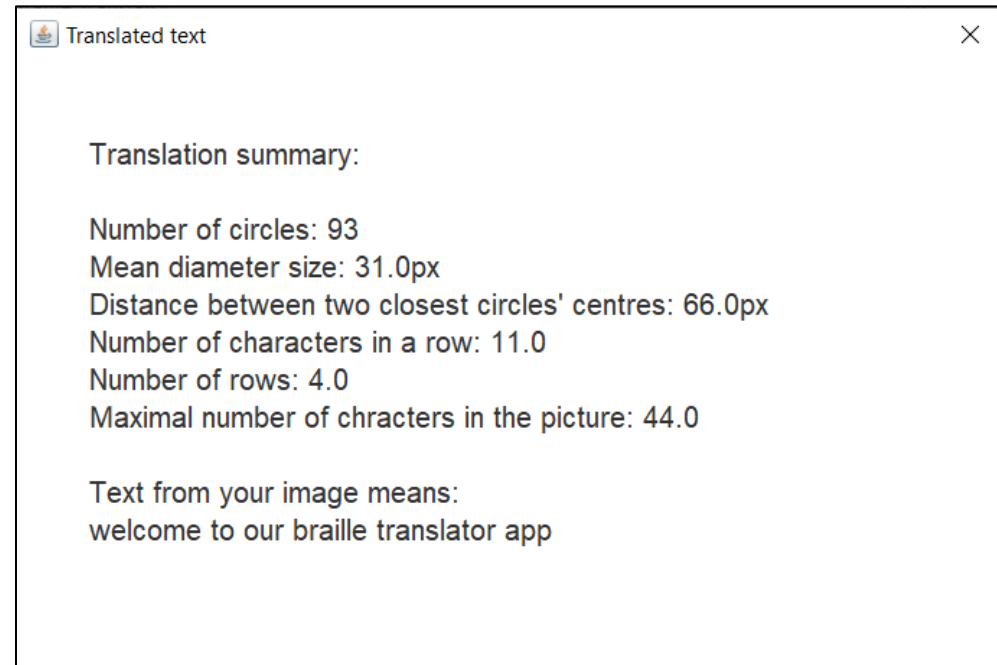
    public static int numofletters = 54;
}
```

Wyświetlanie podsumowania tłumaczenia

W zależności od wyboru użytkownika, podsumowanie tłumaczenia zostanie jedynie wyświetlone w nowym oknie lub dodatkowo zostanie zapisane do pliku.

```
ActionListener actionTranslate = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        if (path!=null) {
            translate=true;
            JDialog mydialog = new JDialog();
            mydialog.setSize(new Dimension(700,600));
            mydialog.setTitle("Translated text");
            mydialog.setModalityType(Dialog.ModalityType.APPLICATION_MODAL); // prevent user from doing something else

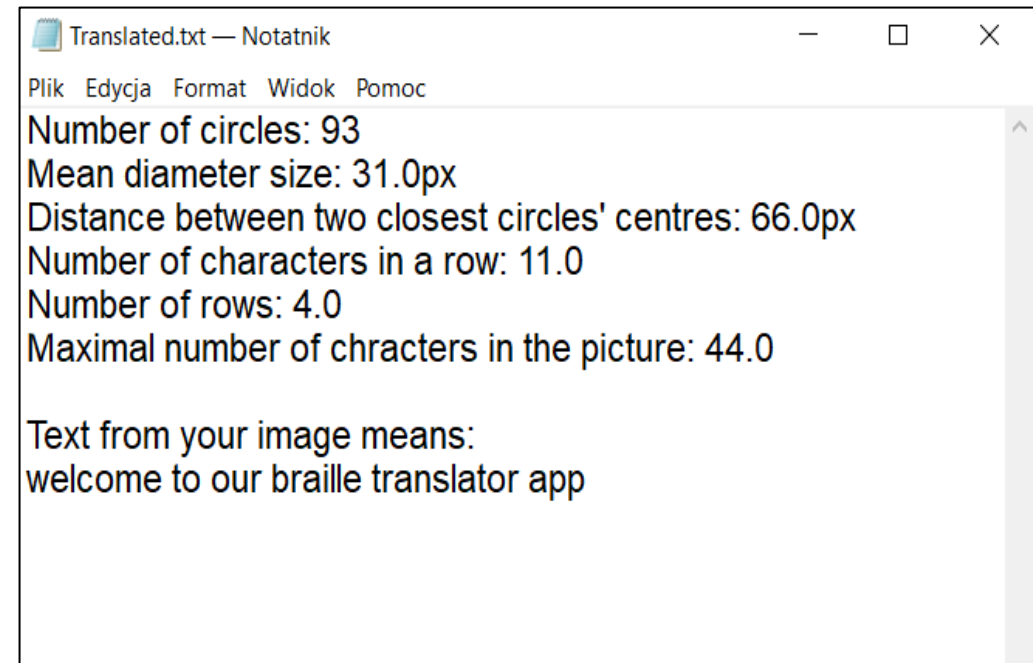
            JTextArea area = new JTextArea("Translation summary: \n\n" + main.getText());
            area.setMargin(new Insets(50,50,50,50));
            area.setFont(area.getFont().deriveFont(20f));
            area.setBounds(10,30, 200,200);
            mydialog.add(area);
            mydialog.setVisible(true);
            //HighGui.imshow("detected circles", main.getImage()); //do usuniecia na koniec?
            //HighGui.waitKey();
            path=null;
        } else {
            JOptionPane.showMessageDialog(null,"You have not choose picture to translate yet!",
                "Warning",JOptionPane.WARNING_MESSAGE);
        }
    }
};
buttonTranslate.addActionListener(actionTranslate);
```



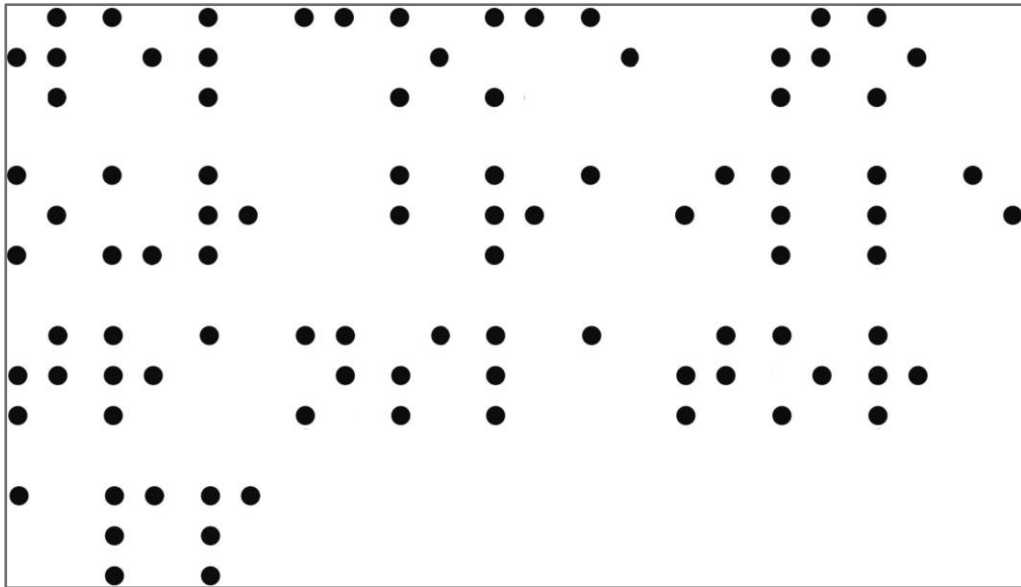
Wyświetlanie podsumowania tłumaczenia

W zależności od wyboru użytkownika, podsumowanie tłumaczenia zostanie jedynie wyświetlone w nowym oknie lub dodatkowo zostanie zapisane do pliku.

```
if (app.getSave() == true) {  
    FileWriter file = new FileWriter("Translated.txt");  
    BufferedWriter translate = new BufferedWriter(file);  
    translate.append(finaltext);  
    translate.append(" ");  
    translate.close();  
}
```



Podsumowanie



Translated text

Translation summary:

Number of circles: 93

Mean diameter size: 31.0px

Distance between two closest circles' centres: 66.0px

Number of characters in a row: 11.0

Number of rows: 4.0

Maximal number of chracters in the picture: 44.0

Text from your image means:

welcome to our braille translator app