

# Systemy Sztucznej Inteligencji

dokumentacja projektu  
"Rozpoznawanie liter i cyfr"

Daniel Piątek | Robert Kwoł, grupa 1b

27 maja 2020

# Część I

## Opis programu

Program służy do narysowania litery bądź cyfry. Do wyboru mamy kilka przycisków do edycji obrazka podczas rysowania. Po wciśnięciu klawisza zapisz i wpisaniu nazwy, następuje zapisanie danej litery do zbioru. Najważniejszym przyciskiem jest porównaj, który sprawdza napisaną literę, bądź cyfrę do tych w zbiorze i wyświetla trzy znaki, pliki oraz procenty prawdopodobieństwa najbardziej pasujących wyników.

## Instrukcja obsługi



### Przyciski:

Długopis - wybiera marker

Gumka - gumuje wybrany obszar

Usuń - całkowicie czyści ekran

Zapisz - po wpisaniu jaką literę chcemy zapisać, program dopisuje narysowany obraz do bazy

Porównaj - porównuje narysowany znak z bazą

### Działanie programu:

Użytkownik rysuje po lewej stronie okna znak który ma zostać rozpoznany. Po narysowaniu znaku należy nacisnąć przycisk "Porównaj", dzięki czemu program przeszuka całą bazę w celu znalezienia najbardziej odpowiednich liter/cyfr. Użytkownik może też zapisać narysowany obraz w celu powiększenia bazy znaków.

## Część II

### Opis działania

Punkty Kluczowe: Pobieramy 2 obrazki, pierwszy to ten narysowany a drugi bierzemy z bazy pokolei. Zapisujemy je jako bitmapy. Następnie tworzymy zmienne, *iloscCzarnych* - ilość pokrywających się narysowanych czarnych pikseli z czarnymi pikselami z bazy, *iloscPrzeciwnych* - ilość pokrywających się czarnych narysowanych pikseli z czarnymi pikselami z bazy, *iloscPrzeciwnych2* - ilość narysowanych białych pikseli z pikseli z czarnymi pikselami z bazy, *sumaCzarnych* - ilość narysowanych czarnych pikseli.

Program na początku sprawdza połowę pikseli z danego obrazu. Jeśli program w małym stopniu odwzorowuje bazę lub jeśli użytkownik wprowadził zbyt małą ilość pikseli to obraz nie jest dalej sprawdzany (w przeciwnym przypadku sprawdzana jest pozostała część obrazu).

Następnie przy użyciu logiki rozmytej przypisane są wagi zmiennym *iloscCzarnych*, *sumaCzarnych* oraz *iloscPrzeciwnych*. Po dopasowaniu wag wszystkim zmiennym, program oblicza końcowe prawdopodobieństwo wystąpienia danego znaku przy pomocy wzoru  
$$\text{wynik} = (\text{iloscCzarnych} - \text{iloscPrzeciwnych} - \text{iloscPrzeciwnych2}) / \text{sumaCzarnych};$$

### Algorytm

```
iloscCzarnych = 0; (ilość narysowanych czarnych oraz czarnych pikseli z bazy)
double iloscPrzeciwnych = 0; (ilość narysowanych czarnych oraz białych pikseli z bazy)
double iloscPrzeciwnych2 = 0; (ilość narysowanych białych pikseli oraz czarnych z bazy)
double sumaCzarnych = 0; (ilość narysowanych czarnych pikseli)
double sumaCzarnychOryginal = 0; (ilość narysowanych czarnych pikseli z bazy)
```

```
for (int Xcount = btm.Width / 2; Xcount < btm.Width; Xcount++)
for (int Ycount = btm.Height / 2; Ycount < btm.Height; Ycount++)
Przeszukuje pierwszą połowę obrazu
```

```
if ((sumaCzarnych < sumaCzarnychOryginal * 1.05 && sumaCzarnych >= sumaCzarnychOryginal * 0.65) || sumaCzarnych < 5000)
```

Porównuje czy ilość narysowanych czarnych pikseli nie jest rozbieżna z ilością czarnych pikseli z bazy, jak również czy użytkownik wprowadził wystarczającą ilość czarnych pikseli

```
if (iloscCzarnych >= sumaCzarnych * 0.8) iloscCzarnych += iloscCzarnych * 0.02;
else if (iloscCzarnych < sumaCzarnych * 0.8 && iloscCzarnych > sumaCzarnych * 0.75) iloscCzarnych += iloscCzarnych * 0.05;
else if (iloscCzarnych <= sumaCzarnych * 0.75 && iloscCzarnych > sumaCzarnych * 0.7) iloscCzarnych += iloscCzarnych * 0.08;
else if (iloscCzarnych <= sumaCzarnych * 0.7 && iloscCzarnych > sumaCzarnych * 0.65) iloscCzarnych += iloscCzarnych * 0.1;
else if (iloscCzarnych <= sumaCzarnych * 0.65 && iloscCzarnych > sumaCzarnych * 0.6) iloscCzarnych += iloscCzarnych * 0.12;
else if (iloscCzarnych <= sumaCzarnych * 0.6 && iloscCzarnych > sumaCzarnych * 0.55) iloscCzarnych += iloscCzarnych * 0.1;
else if (iloscCzarnych <= sumaCzarnych * 0.55 && iloscCzarnych > sumaCzarnych * 0.5) iloscCzarnych += iloscCzarnych * 0.06;
else if (iloscCzarnych <= sumaCzarnych * 0.5 && iloscCzarnych > sumaCzarnych * 0.45) iloscCzarnych += iloscCzarnych * 0.05;
else if (iloscCzarnych <= sumaCzarnych * 0.45 && iloscCzarnych > sumaCzarnych * 0.4) iloscCzarnych += iloscCzarnych
```

```
* 0.04; else if (iloscCzarnych <= sumaCzarnych * 0.4 && iloscCzarnych > sumaCzarnych * 0.35) iloscCzarnych -= iloscCzarnych
* 0.03; else if (iloscCzarnych <= sumaCzarnych * 0.35 && iloscCzarnych > sumaCzarnych * 0.3) iloscCzarnych -= iloscCzarnych
* 0.02; else if (iloscCzarnych <= sumaCzarnych * 0.3) iloscCzarnych -= iloscCzarnych * 0.03;
sumaCzarnych += sumaCzarnych * 0.1;
```

Przypisuje wagi zmiennej zawierającej ilość narysowanych czarnych pikseli

```
if (iloscPrzeciwnych > sumaCzarnych * 0.5) sumaCzarnych += sumaCzarnych * 0.2;
```

Jeśli ilość narysowanych czarnych pikseli i białych pikseli z bazy jest większa, niż połowa ilości narysowanych czarnych pikseli to przypisana jest odpowiednia waga sumie narysowanych czarnych pikseli

```
wynik = (iloscCzarnych - iloscPrzeciwnych - iloscPrzeciwnych2) / sumaCzarnych;
```

Końcowe prawdopodobieństwo obliczamy biorąc dopasowane zmienne i podstawiamy je do danego wzoru

## Implementacja

Implementacja za pomocą pseudokodu ciekawszych algorytmów:

```
input : sender - obiekt przycisku
        e - informacje o stanie i dane zdarzenia skojarzone ze zdarzeniem
        kierowanym
```

```
Button button = (Button)sender
```

```
if (button.Name == "pen" then
```

```
    | MyCanvas.EditingMode = InkCanvasEditingMode.Ink
```

```
end
```

```
if button.Name == "clear" then
```

```
    | MyCanvas.Strokes.Clear()
```

```
end
```

```
if button.Name == "erased" then
```

```
    | MyCanvas.EditingMode = InkCanvasEditingMode.EraseByPoint
```

```
end
```

**Algorithm 1:** Rozpoznawanie 3 przycisków

```

input : sender - obiekt przycisku
        e - informacje o stanie i dane zdarzenia skojarzone ze zdarzeniem
        kierowanym

Save save = new Save()
int width = (int)this.MyCanvas.ActualWidth
int height = (int)this.MyCanvas.ActualHeight
save.zapiszRysunek(savedimage", width, height, MyCanvas)
PunktyKluczowe pkt = new PunktyKluczowe()
IDictionary<string, double> dict = new Dictionary<string, double>()
for fileName in Directory.GetFiles(@"litera") do
    | dict.Add(fileName, pkt.porownajZBaza(fileName))
end
var sortedDict = from entry in dict orderby entry.Value ascending select entry
dict = sortedDict.OrderByDescending(i => i.Value).ToDictionary(i => i.Key, i =>
    i.Value)
textBlock1.Text = dict.Keys.ElementAt(0).Substring(7, 1)
textBlock2.Text = dict.Keys.ElementAt(1).Substring(7, 1)
textBlock3.Text = dict.Keys.ElementAt(2).Substring(7, 1)
textBlockPlik1.Text = dict.Keys.ElementAt(0).Substring(7)
textBlockPlik2.Text = dict.Keys.ElementAt(1).Substring(7)
textBlockPlik3.Text = dict.Keys.ElementAt(2).Substring(7)
textBlockProcent1.Text = dict.Values.ElementAt(0).ToString() + " %"
textBlockProcent2.Text = dict.Values.ElementAt(1).ToString() + " %"
textBlockProcent3.Text = dict.Values.ElementAt(2).ToString() + " %"
Algorithm 2: Funkcje dla przycisku "Porównanie"

```

Do porównania wprowadzonych znaków użyliśmy bazy zdjęć zapisanych w folderze "litery".




Struktura plików w katalogu głównym

Name	Date modified	Type	Size
litery	27.05.2020 17:30	File folder	
litery_1	27.05.2020 17:30	File folder	
paint_test.exe	27.05.2020 17:37	Application	22 KB
paint_test.exe.config	27.05.2020 17:30	XML Configuratio...	1 KB
paint_test.pdb	27.05.2020 17:37	Program Debug D...	56 KB
save.bin	27.05.2020 17:30	BIN File	1 KB
savedimage.jpg	27.05.2020 19:30	JPG File	352 KB

## Testy

- Testy w pełni zakończone sukcesem

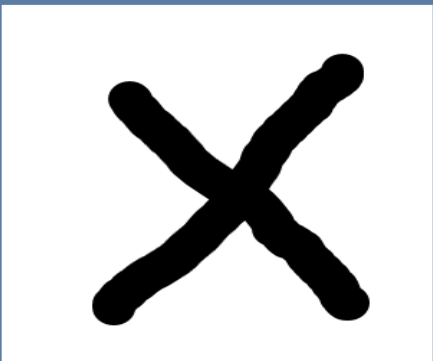
Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł



Znak	Plik	%
W	W8.jpg	87,81 %
U	U7.jpg	70,23 %
V	V5.jpg	65,21 %

Długopis Gumka Usuń Porównaj Zapisz


Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł



Znak	Plik	%
X	X7.jpg	89,58 %
X	X9.jpg	87,4 %
A	A2.jpg	71,73 %

Długopis Gumka Usuń Porównaj Zapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł



Znak	Plik	%
M	M9.jpg	72,51 %
N	N5.jpg	70,66 %
M	M4.jpg	68,1 %

Długopis Gumka Usuń Porównaj Zapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł

Znak	Plik	%
4	43.jpg	70,48 %
4	45.jpg	66,76 %
4	44.jpg	64,4 %

Długopis
Gumka
Usuń
Porównaj
Zapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł

Długopis
Gumka
Usuń
Porównaj
Zapisz

- Testy częściowo zakończone sukcesem

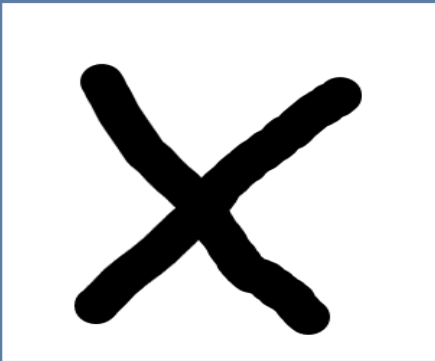
Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł

Znak	Plik	%
6	6.jpg	71,1 %
A	A5.jpg	64,58 %
O	O1.jpg	61,5 %

Długopis
Gumka
Usuń
Porównaj
Zapisz



Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł




Znak	Plik	%
K	K5.jpg	80,56 %
X	X3.jpg	73,69 %
K	K9.jpg	70,54 %

DługopisGumkaUsuń

PorównajZapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł

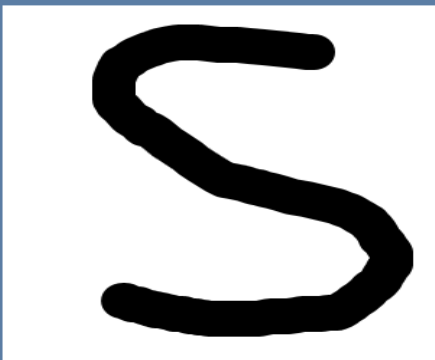


Znak	Plik	%
S	5.jpg	55,35 %
B	B.jpg	55,25 %
S	S.jpg	54,1 %

DługopisGumkaUsuń

PorównajZapisz

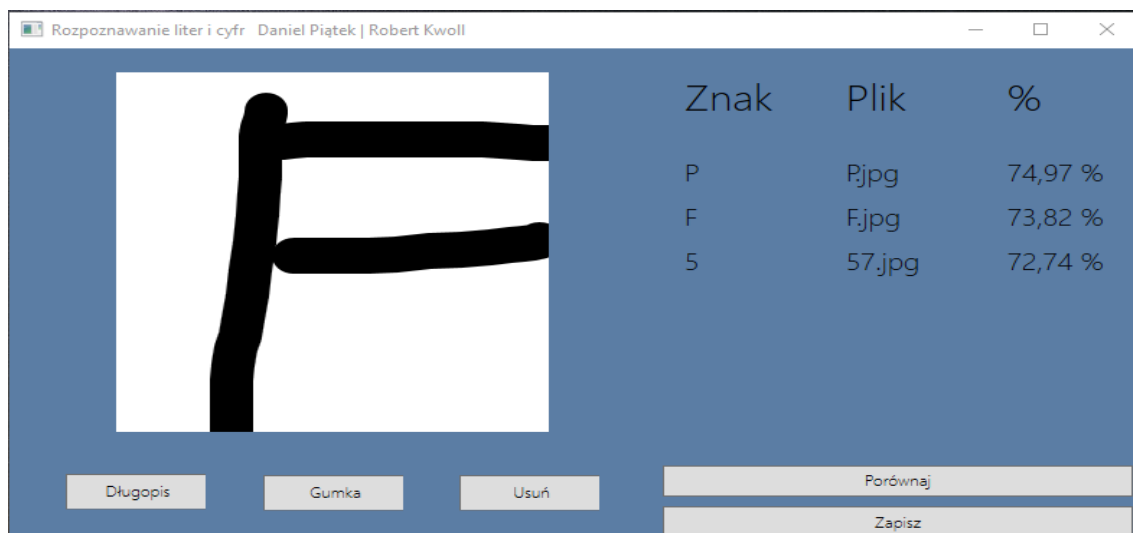
Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł



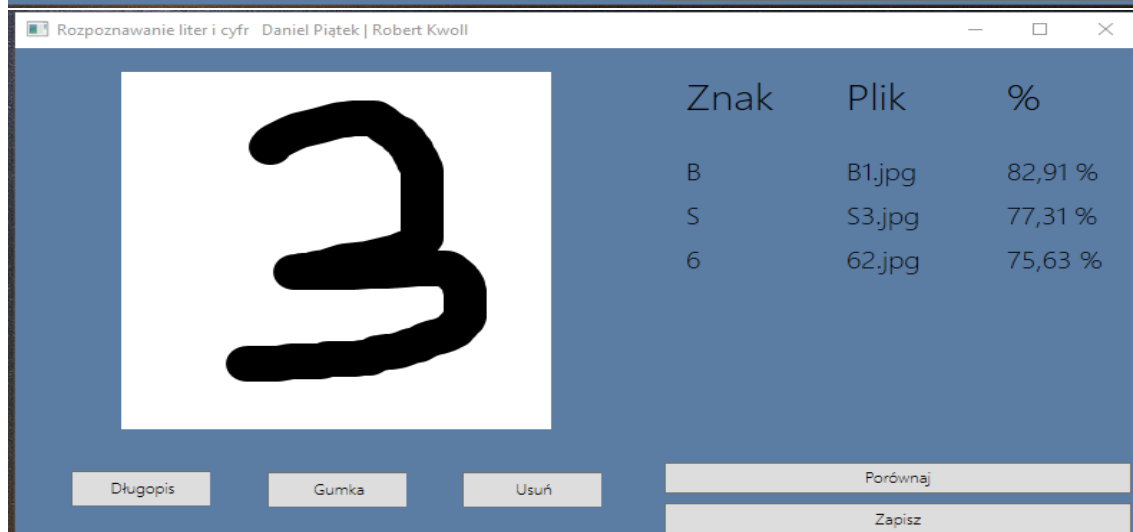
Znak	Plik	%
B	B6.jpg	76,71 %
B	B.jpg	65,71 %
S	S.jpg	64,02 %

DługopisGumkaUsuń


PorównajZapisz



- Testy z błędnym wynikiem



Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł




Znak	Plik	%
B	B5.jpg	72,73 %
D	D3.jpg	62,05 %
H	H7.jpg	54,33 %

Długopis
Gumka
Usuń

Porównaj
Zapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł

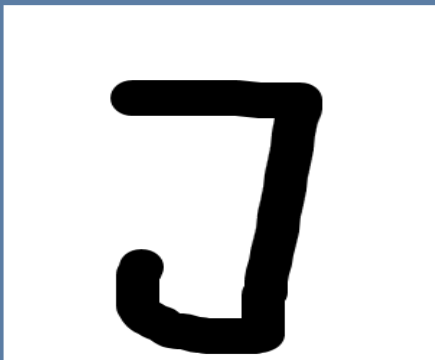


Znak	Plik	%
J	J5.jpg	84,92 %
7	76.jpg	69,96 %
T	T5.jpg	62,73 %

Długopis
Gumka
Usuń

Porównaj
Zapisz

Rozpoznawanie liter i cyfr Daniel Piątek | Robert Kwołł



Znak	Plik	%
A	A1.jpg	55,85 %
9	93.jpg	47,47 %
C	C9.jpg	36,49 %

Długopis
Gumka
Usuń

Porównaj
Zapisz

## **Eksperymenty**

Przeprowadzono test poprawnego działania programu 100 razy. Uzyskano następujące wyniki:

Ilość poprawnych	39%
Częściowo poprawnych	28%
Błędnych	33%

Ilość poprawnych - właściwa litera na pierwszym miejscu

Częściowo poprawnych - właściwa litera na drugim lub trzecim miejscu

Błędnych - Brak wyświetlonej właściwej litery

# Pełen kod aplikacji

## Klasa PunktyKluczowe

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Drawing;
7 using System.Drawing.Imaging;
8 using System.Windows.Controls;
9 using System.IO;
10 using System.Threading;
11 using System.Windows;
12
13 namespace paint_test
14 {
15     class PunktyKluczowe : MainWindow
16     {
17         public PunktyKluczowe()
18         {
19         }
20
21         public double porownajZBaza(string nazwaPlikuZBazy)
22         {
23
24             using (Bitmap btm = new Bitmap(@"savedImage.jpg"))
25             {
26
27                 Bitmap btmF = new Bitmap(nazwaPlikuZBazy);
28                 Bitmap btmTest = new Bitmap(btm.Width, btm.Height);
29
30                 double iloscCzarnych = 0;
31                 double iloscPrzeciwnych = 0;
32                 double iloscPrzeciwnych2 = 0;
33                 double sumaCzarnych = 0;
34                 double sumaCzarnychOryginal = 0;
35                 double wynik = 0;
36
37                 for (int Xcount = 0; Xcount < btm.Width / 2; Xcount++)
38                 {
39                     for (int Ycount = 0; Ycount < btm.Height / 2; Ycount
40                         ++
41                     )
42                     {
43                         if (((btm.GetPixel(Xcount, Ycount).R == 0) && (
44                             btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
45                             GetPixel(Xcount, Ycount).B == 0)) && ((btmF.
46                             GetPixel(Xcount, Ycount).R == 255) && (btmF.
47                             GetPixel(Xcount, Ycount).G == 255) && (btmF.
48                             GetPixel(Xcount, Ycount).B == 255)))
49                             iloscPrzeciwnych++; //narysowany czarny, z
50                             bazy bialy
```

```

44
45         if (((btm.GetPixel(Xcount, Ycount).R == 255) &&
46             (btm.GetPixel(Xcount, Ycount).G == 255) && (
47                 btm.GetPixel(Xcount, Ycount).B == 255)) && ((
48                     btmF.GetPixel(Xcount, Ycount).R == 0) && (
49                         btmF.GetPixel(Xcount, Ycount).G == 0) && (
50                             btmF.GetPixel(Xcount, Ycount).B == 0)))
51             iloscPrzeciwnych2++; //narysowany biały, z
52                                     bazy czarnych
53
54         if (((btm.GetPixel(Xcount, Ycount).R == 0) && (
55             btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
56                 GetPixel(Xcount, Ycount).B == 0)) && ((btmF.
57                     GetPixel(Xcount, Ycount).R == 0) && (btmF.
58                         GetPixel(Xcount, Ycount).G == 0) && (btmF.
59                             GetPixel(Xcount, Ycount).B == 0)))
60             iloscCzarnych++; // narysowany czarny, z
61                                     bazy czarny
62
63         if ((btm.GetPixel(Xcount, Ycount).R == 0) && (
64             btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
65                 GetPixel(Xcount, Ycount).B == 0))
66             sumaCzarnych++;
67
68         if ((btmF.GetPixel(Xcount, Ycount).R == 0) && (
69             btmF.GetPixel(Xcount, Ycount).G == 0) && (
70                 btmF.GetPixel(Xcount, Ycount).B == 0))
71             sumaCzarnychOryginal++;
72     }
73 }
74
75 if ((sumaCzarnych > sumaCzarnychOryginal * 1.05 ||
76     sumaCzarnych < sumaCzarnychOryginal * 0.65) &&
77     sumaCzarnych > 5000)
78     return wynik = 0;
79
80 for (int Xcount = btm.Width / 2; Xcount < btm.Width;
81     Xcount++)
82 {
83     for (int Ycount = btm.Height / 2; Ycount < btm.
84         Height; Ycount++)
85     {
86         if (((btm.GetPixel(Xcount, Ycount).R == 0) && (
87             btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
88                 GetPixel(Xcount, Ycount).B == 0)) && ((btmF.
89                     GetPixel(Xcount, Ycount).R == 255) && (btmF.
90                         GetPixel(Xcount, Ycount).G == 255) && (btmF.
91                             GetPixel(Xcount, Ycount).B == 255)))
92             iloscPrzeciwnych++; //narysowany czarny, z
93                                     bazy biały

```

```

73         if (((btm.GetPixel(Xcount, Ycount).R == 255) &&
              (btm.GetPixel(Xcount, Ycount).G == 255) && (
              btm.GetPixel(Xcount, Ycount).B == 255)) && ((
              btmF.GetPixel(Xcount, Ycount).R == 0) && (
              btmF.GetPixel(Xcount, Ycount).G == 0) && (
              btmF.GetPixel(Xcount, Ycount).B == 0)))
74             iloscPrzeciwnych2++; //narysowany biały, z
              bazy czarnych
75
76
77         if (((btm.GetPixel(Xcount, Ycount).R == 0) && (
              btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
              GetPixel(Xcount, Ycount).B == 0)) && ((btmF.
              GetPixel(Xcount, Ycount).R == 0) && (btmF.
              GetPixel(Xcount, Ycount).G == 0) && (btmF.
              GetPixel(Xcount, Ycount).B == 0)))
78             iloscCzarnych++; // narysowany czarny, z
              bazy czarny
79
80
81         if ((btm.GetPixel(Xcount, Ycount).R == 0) && (
              btm.GetPixel(Xcount, Ycount).G == 0) && (btm.
              GetPixel(Xcount, Ycount).B == 0))
82             sumaCzarnych++;
83
84         if ((btmF.GetPixel(Xcount, Ycount).R == 0) && (
              btmF.GetPixel(Xcount, Ycount).G == 0) && (
              btmF.GetPixel(Xcount, Ycount).B == 0))
85             sumaCzarnychOryginal++;
86     }
87 }
88
89 if ((sumaCzarnych < sumaCzarnychOryginal * 1.05 &&
90     sumaCzarnych >= sumaCzarnychOryginal * 0.65) ||
91     sumaCzarnych < 5000)
92 {
93     iloscCzarnych = (int)(sumaCzarnychOryginal * 0.65);
94     if (iloscCzarnych >= sumaCzarnych * 0.8)
95         iloscCzarnych += iloscCzarnych * 0.02;
96     else if (iloscCzarnych < sumaCzarnych * 0.8 &&
97         iloscCzarnych > sumaCzarnych * 0.75)
98         iloscCzarnych += iloscCzarnych * 0.05;
99     else if (iloscCzarnych <= sumaCzarnych * 0.75 &&
100         iloscCzarnych > sumaCzarnych * 0.7) iloscCzarnych
101         += iloscCzarnych * 0.08;
102     else if (iloscCzarnych <= sumaCzarnych * 0.7 &&
103         iloscCzarnych > sumaCzarnych * 0.65)
104         iloscCzarnych += iloscCzarnych * 0.1;
105     else if (iloscCzarnych <= sumaCzarnych * 0.65 &&
106         iloscCzarnych > sumaCzarnych * 0.6) iloscCzarnych
107         += iloscCzarnych * 0.12;
108     else if (iloscCzarnych <= sumaCzarnych * 0.6 &&
109         iloscCzarnych > sumaCzarnych * 0.55)
110         iloscCzarnych += iloscCzarnych * 0.1;

```

```

98     else if (iloscCzarnych <= sumaCzarnych * 0.55 &&
          iloscCzarnych > sumaCzarnych * 0.5) iloscCzarnych
          += iloscCzarnych * 0.06;
99     else if (iloscCzarnych <= sumaCzarnych * 0.5 &&
          iloscCzarnych > sumaCzarnych * 0.45)
          iloscCzarnych += iloscCzarnych * 0.05;
100    else if (iloscCzarnych <= sumaCzarnych * 0.45 &&
          iloscCzarnych > sumaCzarnych * 0.4) iloscCzarnych
          += iloscCzarnych * 0.04;
101    else if (iloscCzarnych <= sumaCzarnych * 0.4 &&
          iloscCzarnych > sumaCzarnych * 0.35)
          iloscCzarnych -= iloscCzarnych * 0.03;
102    else if (iloscCzarnych <= sumaCzarnych * 0.35 &&
          iloscCzarnych > sumaCzarnych * 0.3) iloscCzarnych
          -= iloscCzarnych * 0.02;
103    else if (iloscCzarnych <= sumaCzarnych * 0.3)
104    {
105        iloscCzarnych -= iloscCzarnych * 0.03;
106        sumaCzarnych += sumaCzarnych * 0.1;
107    }
108
109    //iloscPrzeciwnych
110    if (iloscPrzeciwnych <= sumaCzarnych * 0.2)
          iloscPrzeciwnych /= 4;
111    else if (iloscPrzeciwnych > sumaCzarnych * 0.2 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.25)
          iloscPrzeciwnych /= 3;
112    else if (iloscPrzeciwnych > sumaCzarnych * 0.25 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.3)
          iloscPrzeciwnych /= 2.5 - iloscPrzeciwnych *
          0.02;
113    else if (iloscPrzeciwnych > sumaCzarnych * 0.3 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.35)
          iloscPrzeciwnych /= 2.3 + iloscPrzeciwnych * 0.2;
114    else if (iloscPrzeciwnych > sumaCzarnych * 0.35 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.4)
          iloscPrzeciwnych /= 2.2 + iloscPrzeciwnych *
          0.25;
115    else if (iloscPrzeciwnych > sumaCzarnych * 0.4 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.45)
          iloscPrzeciwnych /= 2 + iloscPrzeciwnych * 0.3;
116    else if (iloscPrzeciwnych > sumaCzarnych * 0.45 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.5)
          iloscPrzeciwnych /= 1.8 + iloscPrzeciwnych *
          0.35;
117    else if (iloscPrzeciwnych > sumaCzarnych * 0.5 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.55)
          iloscPrzeciwnych /= 1.6 + iloscPrzeciwnych * 0.4;
118    else if (iloscPrzeciwnych > sumaCzarnych * 0.55 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.6)
          iloscPrzeciwnych /= 1.5 + iloscPrzeciwnych *
          0.45;
119    else if (iloscPrzeciwnych > sumaCzarnych * 0.6 &&
          iloscPrzeciwnych <= sumaCzarnych * 0.65)
          iloscPrzeciwnych += (iloscPrzeciwnych * 0.2) /

```



```

120         1.5;
121     else if (iloscPrzeciwnych > sumaCzarnych * 0.65)
122         iloscPrzeciwnych += (iloscPrzeciwnych * 0.4) / 2;
123
124     if (iloscPrzeciwnych2 <= sumaCzarnych * 0.2)
125         iloscPrzeciwnych2 /= 4;
126     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.2 &&
127         iloscPrzeciwnych2 <= sumaCzarnych * 0.25)
128         iloscPrzeciwnych2 /= 3;
129     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.25 &&
130         iloscPrzeciwnych2 <= sumaCzarnych * 0.3)
131         iloscPrzeciwnych2 /= 2.5 - iloscPrzeciwnych2 *
132             0.02;
133     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.3 &&
134         iloscPrzeciwnych2 <= sumaCzarnych * 0.35)
135         iloscPrzeciwnych2 /= 2.3 + iloscPrzeciwnych2 *
136             0.2;
137     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.35 &&
138         iloscPrzeciwnych2 <= sumaCzarnych * 0.4)
139         iloscPrzeciwnych2 /= 2.1 + iloscPrzeciwnych2 *
140             0.25;
141     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.4 &&
142         iloscPrzeciwnych2 <= sumaCzarnych * 0.45)
143         iloscPrzeciwnych2 /= 2 + iloscPrzeciwnych2 * 0.3;
144     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.45 &&
145         iloscPrzeciwnych2 <= sumaCzarnych * 0.5)
146         iloscPrzeciwnych2 /= 1.8 + iloscPrzeciwnych2 *
147             0.35;
148     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.5 &&
149         iloscPrzeciwnych2 <= sumaCzarnych * 0.55)
150         iloscPrzeciwnych2 /= 1.6 + iloscPrzeciwnych2 *
151             0.4;
152     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.55 &&
153         iloscPrzeciwnych2 <= sumaCzarnych * 0.6)
154         iloscPrzeciwnych2 /= 1.5 + iloscPrzeciwnych2 *
155             0.45;
156     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.6 &&
157         iloscPrzeciwnych2 <= sumaCzarnych * 0.65)
158         iloscPrzeciwnych2 += (iloscPrzeciwnych2 * 0.2) /
159             1.5;
160     else if (iloscPrzeciwnych2 > sumaCzarnych * 0.65)
161         iloscPrzeciwnych2 += (iloscPrzeciwnych2 * 0.4) /
162             2;
163
164     if (iloscPrzeciwnych > sumaCzarnych * 0.5)
165         sumaCzarnych += sumaCzarnych * 0.2;
166
167     wynik = (iloscCzarnych - iloscPrzeciwnych -
168         iloscPrzeciwnych2) / sumaCzarnych;
169 }
170
171 else
172     wynik = 0;
173
174
175

```

```

143         if (wynik < 0)
144             wynik = 0;
145
146
147         wynik *= 100;
148         return (Math.Round(wynik, 2));
149
150     }
151 }
152
153 }
154 }

```

---

## Główna klasa

```

1 using Microsoft.Win32;
2 using System;
3 using System.Collections.Generic;
4 using System.Drawing;
5 using System.IO;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows;
10 using System.Windows.Controls;
11 using System.Windows.Data;
12 using System.Windows.Documents;
13 using System.Windows.Ink;
14 using System.Windows.Input;
15 using System.Windows.Media;
16 using System.Windows.Media.Imaging;
17 using System.Windows.Navigation;
18 using System.Windows.Shapes;
19 using Microsoft.VisualBasic;
20
21 namespace paint_test
22 {
23     /// <summary>
24     /// Logika interakcji dla klasy MainWindow.xaml
25     /// </summary>
26     public partial class MainWindow : Window
27     {
28         public MainWindow()
29         {
30             InitializeComponent();
31             MyCanvas.DefaultDrawingAttributes.Width = 30;
32             MyCanvas.DefaultDrawingAttributes.Height = 30;
33         }
34
35         private void paint_Click(object sender, RoutedEventArgs e)
36         {
37             Button button = (Button)sender;
38
39             if (button.Name == "pen")

```

```

40         MyCanvas.EditingMode = InkCanvasEditingMode.Ink;
41
42         if (button.Name == "clear")
43             MyCanvas.Strokes.Clear();
44
45         if (button.Name == "erased")
46             MyCanvas.EditingMode = InkCanvasEditingMode.EraseByPoint
47             ;
48     }
49
50     private void porownaj_Click(object sender, RoutedEventArgs e)
51     {
52         Save save = new Save();
53         int width = (int)this.MyCanvas.ActualWidth;
54         int height = (int)this.MyCanvas.ActualHeight;
55         save.zapiszRysunek("savedimage", width, height, MyCanvas);
56
57         PunktyKluczowe pkt = new PunktyKluczowe();
58         IDictionary<string, double> dict = new Dictionary<string,
59             double>();
60
61         foreach (string fileName in Directory.GetFiles(@"liter\"))
62         {
63             dict.Add(fileName, pkt.porownajZBaza(fileName));
64         }
65
66         var sortedDict = from entry in dict orderby entry.Value
67             ascending select entry;
68         dict = sortedDict.OrderByDescending(i => i.Value).
69             ToDictionary(i => i.Key, i => i.Value);
70
71         textBlock1.Text = dict.Keys.ElementAt(0).Substring(7, 1);
72         textBlock2.Text = dict.Keys.ElementAt(1).Substring(7, 1);
73         textBlock3.Text = dict.Keys.ElementAt(2).Substring(7, 1);
74
75         textBlockPlik1.Text = dict.Keys.ElementAt(0).Substring(7);
76         textBlockPlik2.Text = dict.Keys.ElementAt(1).Substring(7);
77         textBlockPlik3.Text = dict.Keys.ElementAt(2).Substring(7);
78
79         textBlockProcent1.Text = dict.Values.ElementAt(0).ToString()
80             + " % ";
81         textBlockProcent2.Text = dict.Values.ElementAt(1).ToString()
82             + " % ";
83         textBlockProcent3.Text = dict.Values.ElementAt(2).ToString()
84             + " % ";
85     }
86
87     private void zapisz_Click(object sender, RoutedEventArgs e)
88     {
89         InputBox.Visibility = System.Windows.Visibility.Visible;
90     }
91
92     private void YesButton_Click(object sender, RoutedEventArgs e)

```

```

88     {
89         InputBox.Visibility = System.Windows.Visibility.Collapsed;
90         String input = InputTextBox.Text;
91         Save save = new Save();
92         int width = (int)this.MyCanvas.ActualWidth;
93         int height = (int)this.MyCanvas.ActualHeight;
94         save.zapiszRysunek(input, width, height, MyCanvas);
95         InputTextBox.Text = String.Empty;
96     }
97
98
99     private void NoButton_Click(object sender, RoutedEventArgs e)
100    {
101        InputBox.Visibility = System.Windows.Visibility.Collapsed;
102        InputTextBox.Text = String.Empty;
103    }
104
105    }
106 }

```

---

## XAML

```

1
2 <Window x:Class="paint_test.MainWindow"
3         xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation
4         xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5         xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6         xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility
7         /2006"
8         xmlns:local="clr-namespace:paint_test"
9         mc:Ignorable="d"
10        Title="Rozpoznawanie liter i cyfr Daniel Piatek | Robert Kwoll
11        " Height="450" Width="800" FontFamily="Segoe UI Light"
12        MinHeight="450" MinWidth="800">
13    <Grid>
14        <Grid.ColumnDefinitions>
15            <ColumnDefinition Width="4*" />
16            <ColumnDefinition Width="1*" />
17            <ColumnDefinition Width="1*" />
18            <ColumnDefinition Width="1*" />
19        </Grid.ColumnDefinitions>
20
21        <Grid.RowDefinitions>
22            <RowDefinition Height="1*" />
23            <RowDefinition Height="9*" />
24            <RowDefinition Height="1*" />
25            <RowDefinition Height="1*" />
26        </Grid.RowDefinitions>
27
28        <StackPanel Grid.Column="0" Grid.Row="0" Grid.RowSpan="2"
29            Background="#5b7da4">

```

```

29         <Grid Height="300" Width="300" Margin="20">
30             <InkCanvas x:Name="MyCanvas" Height="300" Width="300"
31                 Background="white"/>
32         </Grid>
33     </StackPanel>
34     <StackPanel Grid.Row="2" Grid.RowSpan="2" Background="#5b7da4"
35         Margin="-1">
36         <Grid>
37             <Button x:Name="clear" Content="K[U+FFFD] Click="
38                 paint_Click" Width="100" Height="30" Margin="
39                 313,14,40,-14"/>
40         </Grid>
41         <Grid>
42             <Button x:Name="erased" Content="Gumka" Click="
43                 paint_Click" Margin="177,-16,176,24" Width="100"
44                 Height="30"/>
45         </Grid>
46         <Grid>
47             <Button x:Name="pen" Content="D[U+FFFD] Click="
48                 paint_Click" Margin="40,-55,313,63" Width="100"
49                 Height="30"/>
50         </Grid>
51     </StackPanel>
52
53     <StackPanel Background="#5b7da4" Grid.Column="1" Grid.Row="2"
54         Grid.ColumnSpan="3" Margin="-1">
55         <Button x:Name="porownaj" Content="P[U+FFFD] Click="
56             porownaj_Click" Margin="6" Height="26"/>
57     </StackPanel>
58     <StackPanel Background="#5b7da4" Grid.Column="1" Grid.Row="3"
59         Grid.ColumnSpan="3" Margin="-1">
60         <Button x:Name="zapiszPlik" Content="Zapisz" Click="
61             zapisz_Click" Margin="6" Height="26"/>
62     </StackPanel>
63
64     <StackPanel Grid.Column="1" Grid.Row="0" Grid.RowSpan="2"
65         Background="#5b7da4">
66         <TextBlock x:Name="znak" Margin="20,20,0,10" Text="Znak"
67             Grid.Column="3" FontSize="30"></TextBlock>
68         <TextBlock x:Name="textBlock1" Margin="20,20,0,10" Text=""
69             FontSize="20"></TextBlock>
70         <TextBlock x:Name="textBlock2" Margin="20,0,0,10" Text=""
71             FontSize="20"></TextBlock>
72         <TextBlock x:Name="textBlock3" Margin="20,0,0,10" Text=""
73             FontSize="20"></TextBlock>
74     </StackPanel>
75
76     <StackPanel Grid.Column="2" Grid.Row="0" Grid.RowSpan="2"
77         Background="#5b7da4">
78         <TextBlock x:Name="plik" Margin="20,20,0,10" Text="Plik"
79             Grid.Column="3" FontSize="30"></TextBlock>
80         <TextBlock x:Name="textBlockPlik1" Margin="20,20,0,10" Text=""

```

```

        ="" FontSize="20"></TextBlock>
65     <TextBlock x:Name="textBlockPlik2" Margin="20,0,0,10" Text="
        " FontSize="20"></TextBlock>
66     <TextBlock x:Name="textBlockPlik3" Margin="20,0,0,10" Text="
        " FontSize="20"></TextBlock>
67 </StackPanel>
68
69 <StackPanel Grid.Column="3" Grid.Row="0" Grid.RowSpan="2"
    Background="#5b7da4">
70     <TextBlock x:Name="procent" Margin="20,20,0,10" Text="%"
        Grid.Column="3" FontSize="30" ></TextBlock>
71     <TextBlock x:Name="textBlockProcent1" Margin="20,20,0,10"
        Text="" FontSize="20"></TextBlock>
72     <TextBlock x:Name="textBlockProcent2" Margin="20,0,0,10"
        Text="" FontSize="20" ></TextBlock>
73     <TextBlock x:Name="textBlockProcent3" Margin="20,0,0,10"
        Text="" FontSize="20"></TextBlock>
74 </StackPanel>
75
76
77
78
79
80 <Grid x:Name="InputBox" Visibility="Collapsed" Grid.Row="0"
    Grid.RowSpan="2">
81     <Grid Background="Black" Opacity="0.5"/>
82     <Border
83         MinWidth="250"
84         BorderBrush="Black"
85         BorderThickness="1"
86         HorizontalAlignment="Center"
87         VerticalAlignment="Center">
88         <StackPanel>
89             <TextBlock Margin="5" Text="Wzrost: [U+FFFD] [U+FFFD] [U+FFFD]
                zapisujesz?:" FontWeight="Bold" FontFamily="
                Cambria" Background=" #4ab595" />
90             <TextBox MinWidth="150" HorizontalAlignment="Center"
                VerticalAlignment="Center" x:Name="InputTextBox"
                />
91             <StackPanel Orientation="Horizontal"
                HorizontalAlignment="Center">
92                 <Button x:Name="YesButton" Margin="5" Content="
                    Zapisz" Background=" #4ab595" Click="
                    YesButton_Click"/>
93                 <Button x:Name="NoButton" Margin="5" Content="
                    Wzrost: [U+FFFD] Background=" #4ab595" Click="
                    NoButton_Click" />
94             </StackPanel>
95         </StackPanel>
96     </Border>
97 </Grid>
98
99
100
101

```

```
102     </Grid>
103 </Window>
```

---

## Klasa Save

```
1 using Microsoft.Win32;
2 using System;
3 using System.Collections.Generic;
4 using System.IO;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8 using System.Windows.Media;
9 using System.Windows.Media.Imaging;
10
11 namespace paint_test
12 {
13     class Save
14     {
15
16         public void zapiszRysunek(String nazwaPliku,int width,int height
17             , System.Windows.Media.Visual MyCanvas)
18         {
19             SaveFileDialog dlg = new SaveFileDialog();
20             dlg.FileName = "savedimage.jpg";
21             int krotnoscPliku = 0;
22             string filename;
23
24             if (nazwaPliku != "savedimage")
25             {
26                 if (File.Exists(@"liter\ " + nazwaPliku + ".jpg"))
27                     do
28                     {
29                         krotnoscPliku++;
30                     } while (File.Exists(@"liter\ " + nazwaPliku +
31                         krotnoscPliku + ".jpg")); // sprawdza czy plik o
32                         danej nazwie istnieje
33
34                 filename = @"liter\ " + nazwaPliku + krotnoscPliku + ".
35                     jpg";
36             }
37             else
38                 filename = dlg.FileName;
39
40             RenderTargetBitmap rtb =
41                 new RenderTargetBitmap(width, height, 96d, 96d, PixelFormats
42                     .Default);
43             rtb.Render(MyCanvas);
44
45             using (FileStream fs = new FileStream(filename, FileMode.
46                 Create))
47             {
48                 File.SetAttributes(filename, FileAttributes.Normal);
49                 BmpBitmapEncoder encoder = new BmpBitmapEncoder();
```

```
44         encoder.Frames.Add(BitmapFrame.Create(rtb));
45         encoder.Save(fs);
46         fs.Close();
47     }
48
49 }
50
51 }
```

---