

# Przetwarzanie i analiza obrazu

Ewa Tymoszevska gr 77

Projekt 2.

Implementacja zadania odczytania wyrazow z obrazu planszy do gry Scrabble i odnalezienia wyrazow nieobecnych na planszy z listy slow.

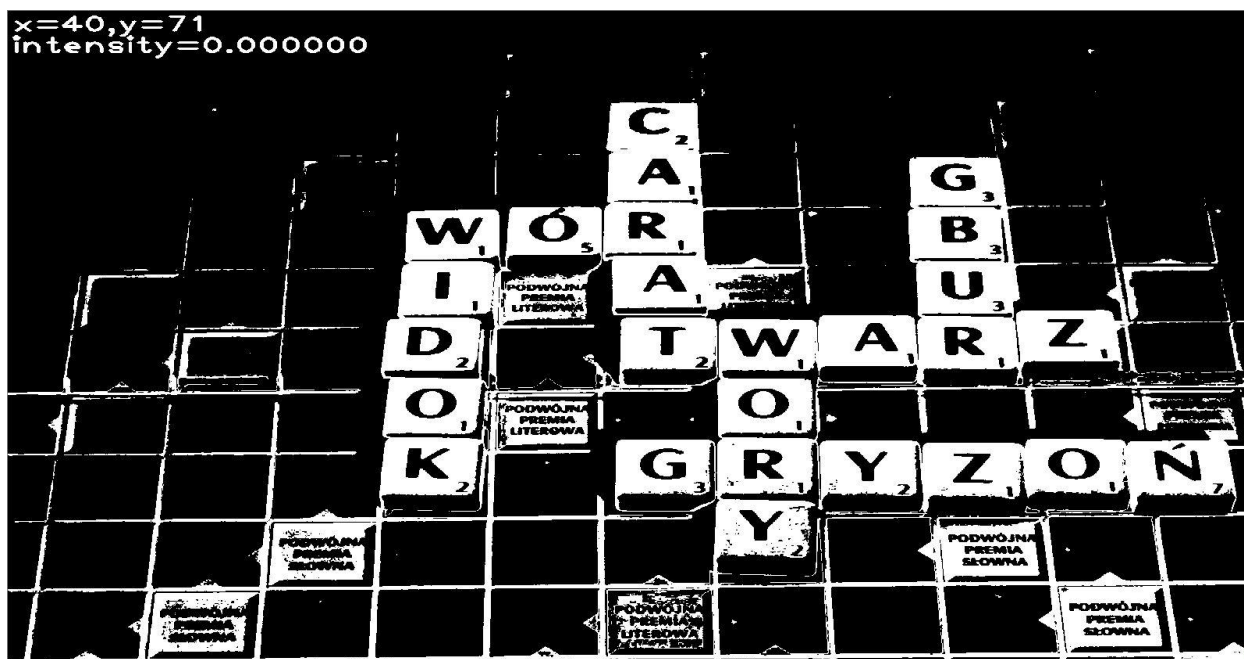
## 1. Przygotowanie obrazu

Zadanie rozwiązano przez napisanie programu w języku C++ w środowisku Visual Studio Professional 2013. Do programu wczytano oryginalny obraz.



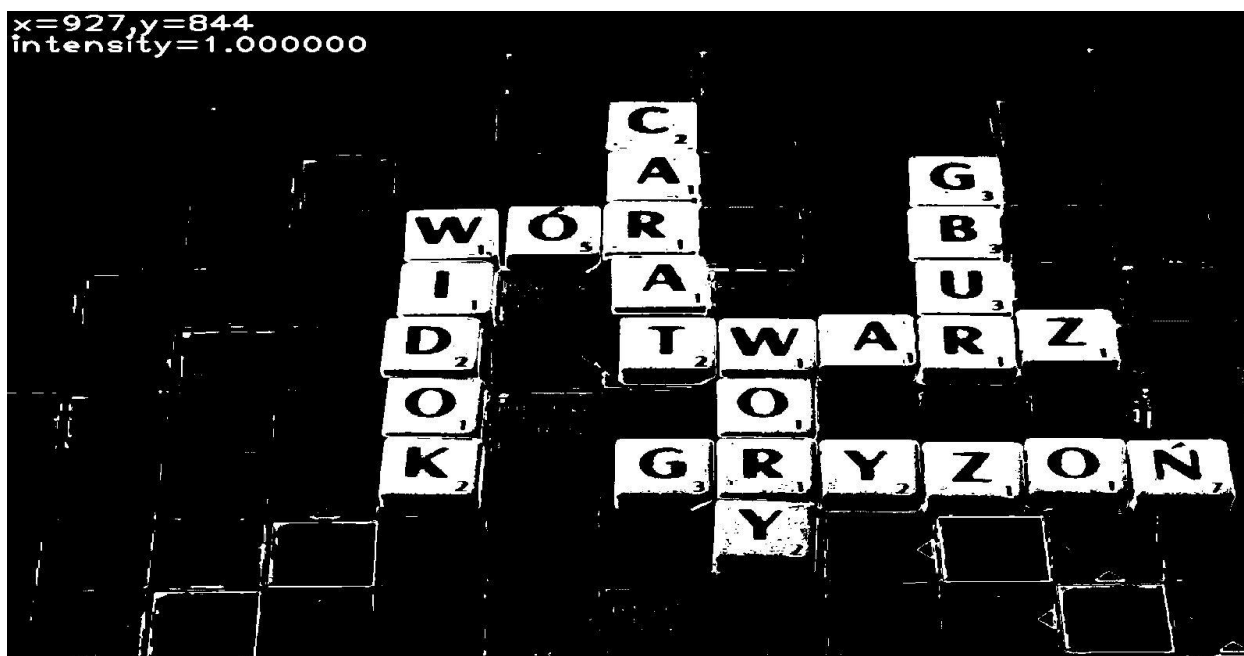
Rys. 1 Obraz Oryginalny

Następnie stworzono obraz binarny do późniejszego użytku.

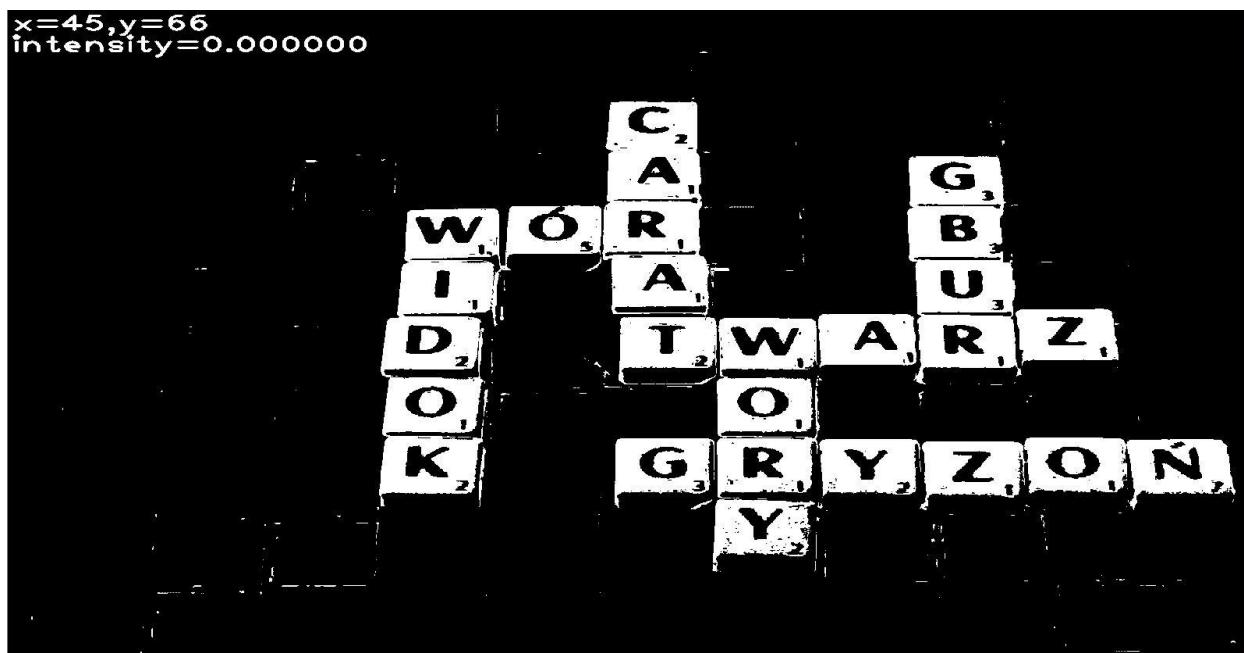


Rys 2. Obraz binarny

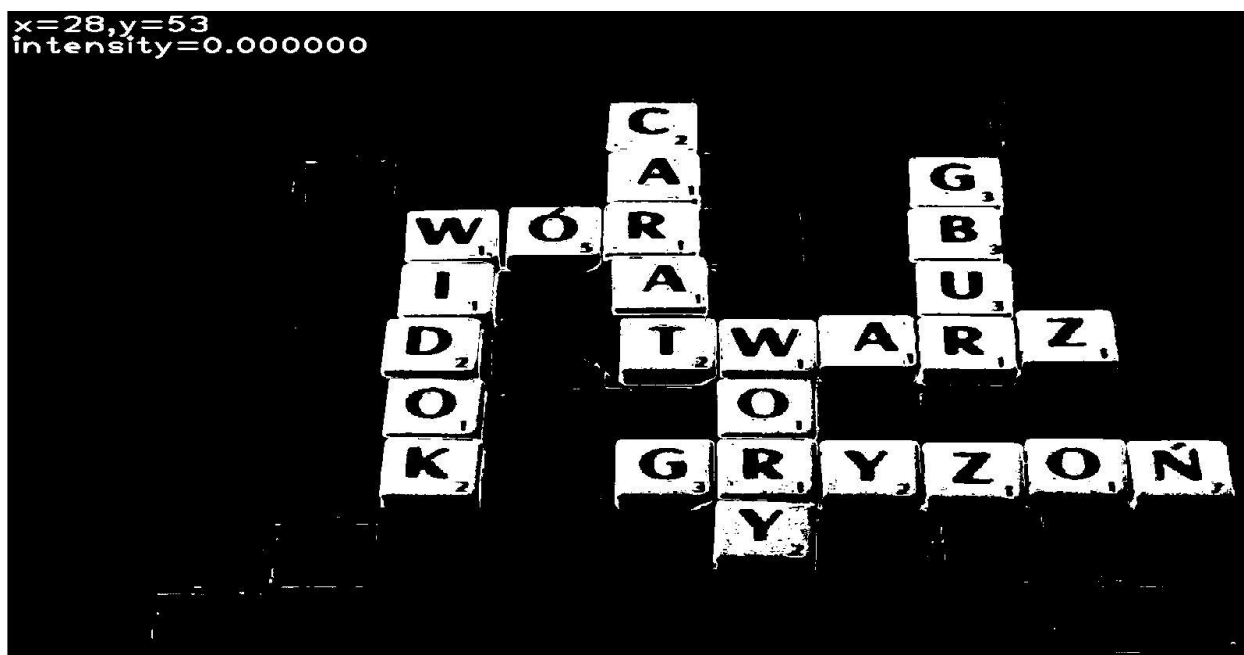
W celu usunięcia szumu obliczono zakresy kolorów czerwonego, niebieskiego i zielonego które można zaobserwować na najjaśniejszym i najciemniejszym kwadraciku. Dzięki temu uzyskano proporcje kolorów czerwonego do zielonego, niebieskiego do zielonego i czerwonego do niebieskiego. Na obrazie binarnym usunięto kolory pozostające poza zakresami tych proporcji dla ciemnego i jasnego kwadracika na obrazie kolorowym.



Rys 3. Obraz po usunięciu pikseli poza zakresem proporcji czerwonego do zielonego



Rys 4. Obraz po usunięciu pikseli poza zakresem proporcji niebieskiego do zielonego



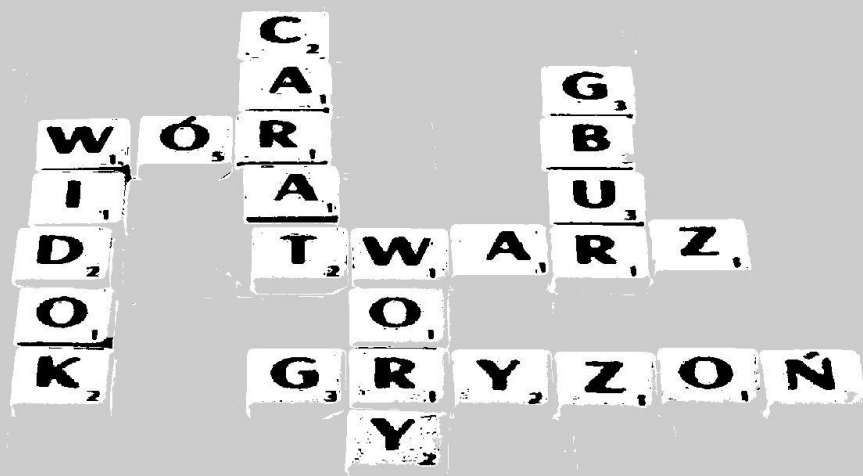
Rys 5. Obraz po usunięciu pikseli poza zakresem proporcji czerwonego do niebieskiego

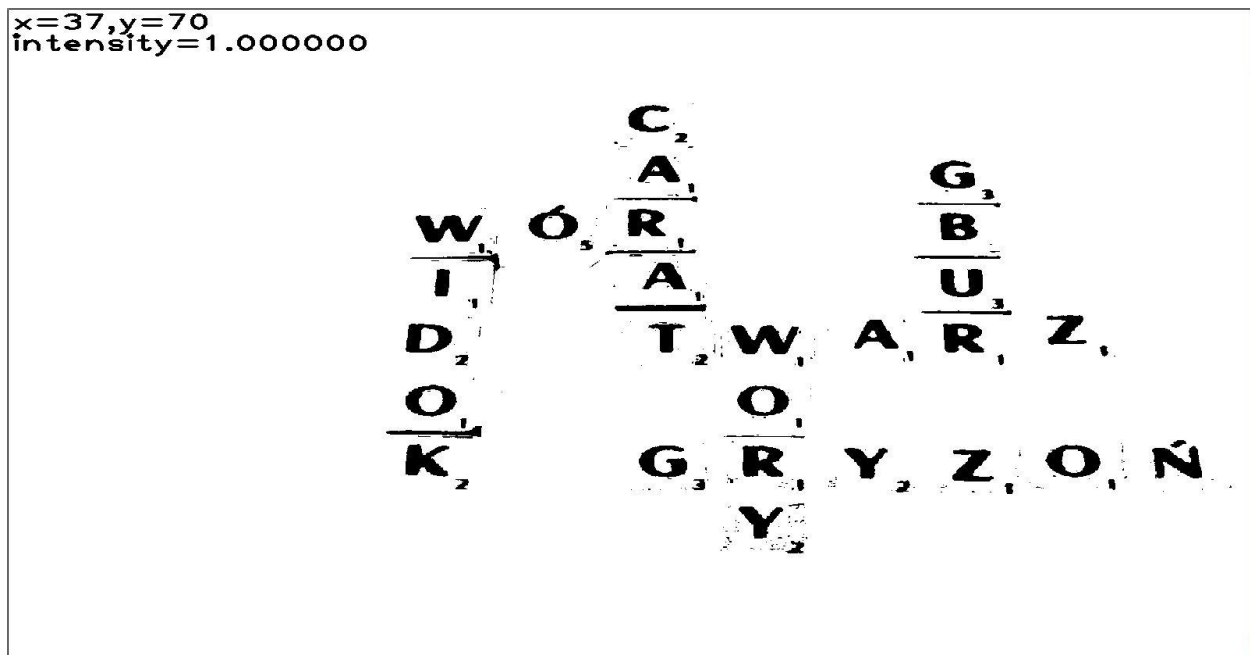
Przy użyciu algorytmu podobnego do algorytmu segmentacji przedstawionego na zajęciach Przetwarzania i Analizy Obrazu usunięto tło obrazu. Oznaczono piksel (0;0) intensywnością 0.8. Następnie na obrazie odnajdowano wszystkie piksele o intensywności 0.8 które sąsiadowały z pikselami o intensywności 0.8. Usunięcie tła pozwoliło na pozbycie się znacznych ilości szumu.

x=51,y=70  
intensity=0.800000



x=39,y=44  
intensity=0.800000

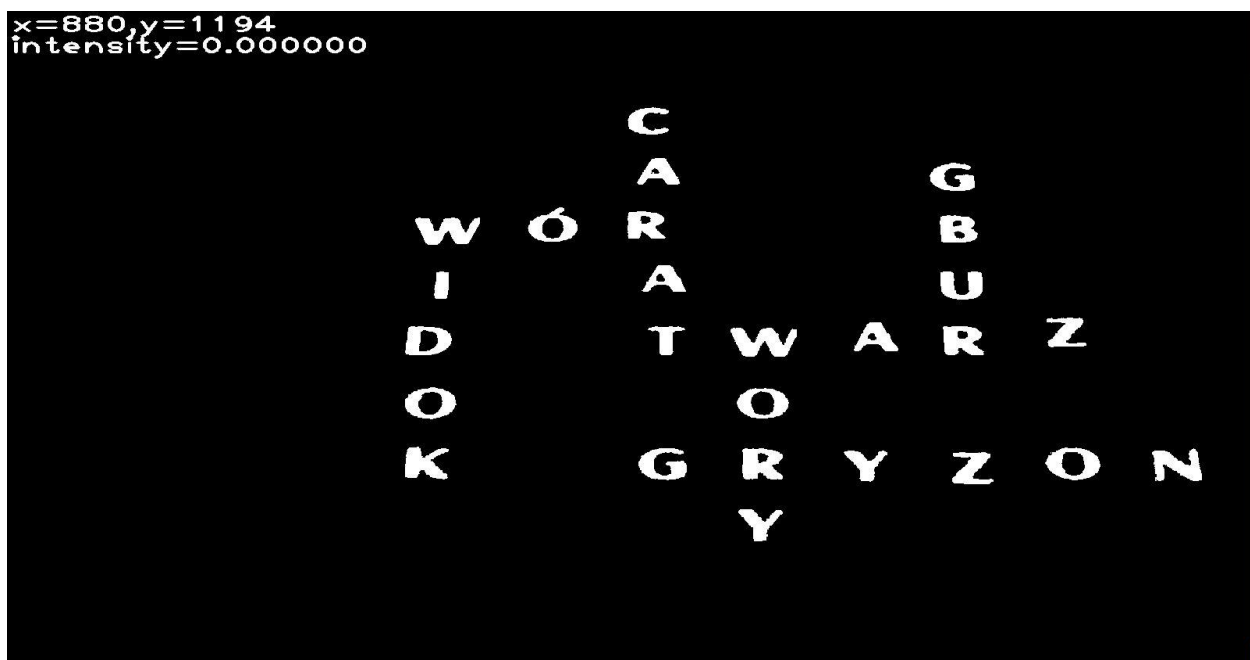




Rys 6, 7, 8. Etapy usunięcia tła

Wtedy można już było dokonać właściwej segmentacji. Wczytano obraz począwszy od piksela (0,0) i podążając w prawo i w dół. Każdemu czarnemu pikselowi przyznano unikalną liczbę całkowitą. Podążając w prawo i w dół każdemu pikselowi przyznano liczbę najniższą z jego 8-pikselowego sąsiedztwa. Operacje powtórzono czytając obraz od dołu i w lewo.

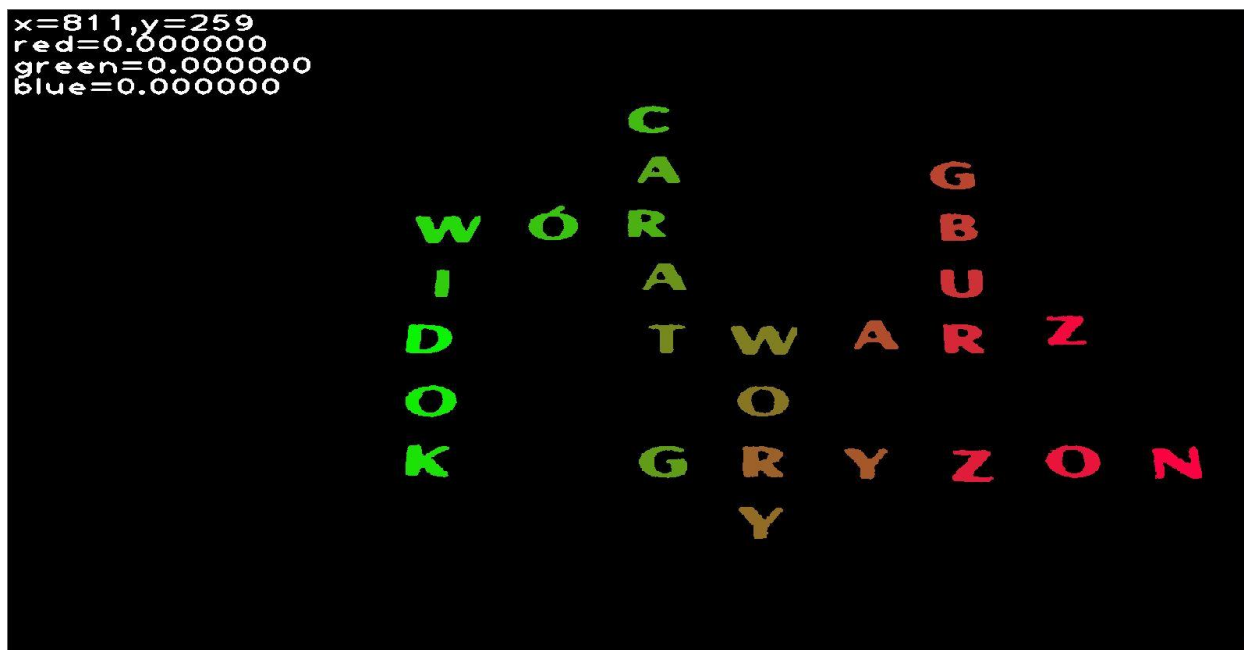
Z obrazu usunięto segmenty dłuższe niż 100 pikseli i te o polu powierzchni mniejszym niż 800 pikseli dzięki czemu ostatecznie oczyszczono obraz z szumu.



Rys 9. Obraz oczyszczony z szumu

Segmenty policzono. Wyznaczono piksele najbardziej wysunięte na lewo, prawo, do góry i do dołu dla każdego segment. W ten sposób otrzymano kwadraty w których zamykają się segmenty.

Dla wizualizacji segmentacji pokolorowano każdy segment odcieniami kolorów RGB proporcjonalnymi dla numeru segmentu.

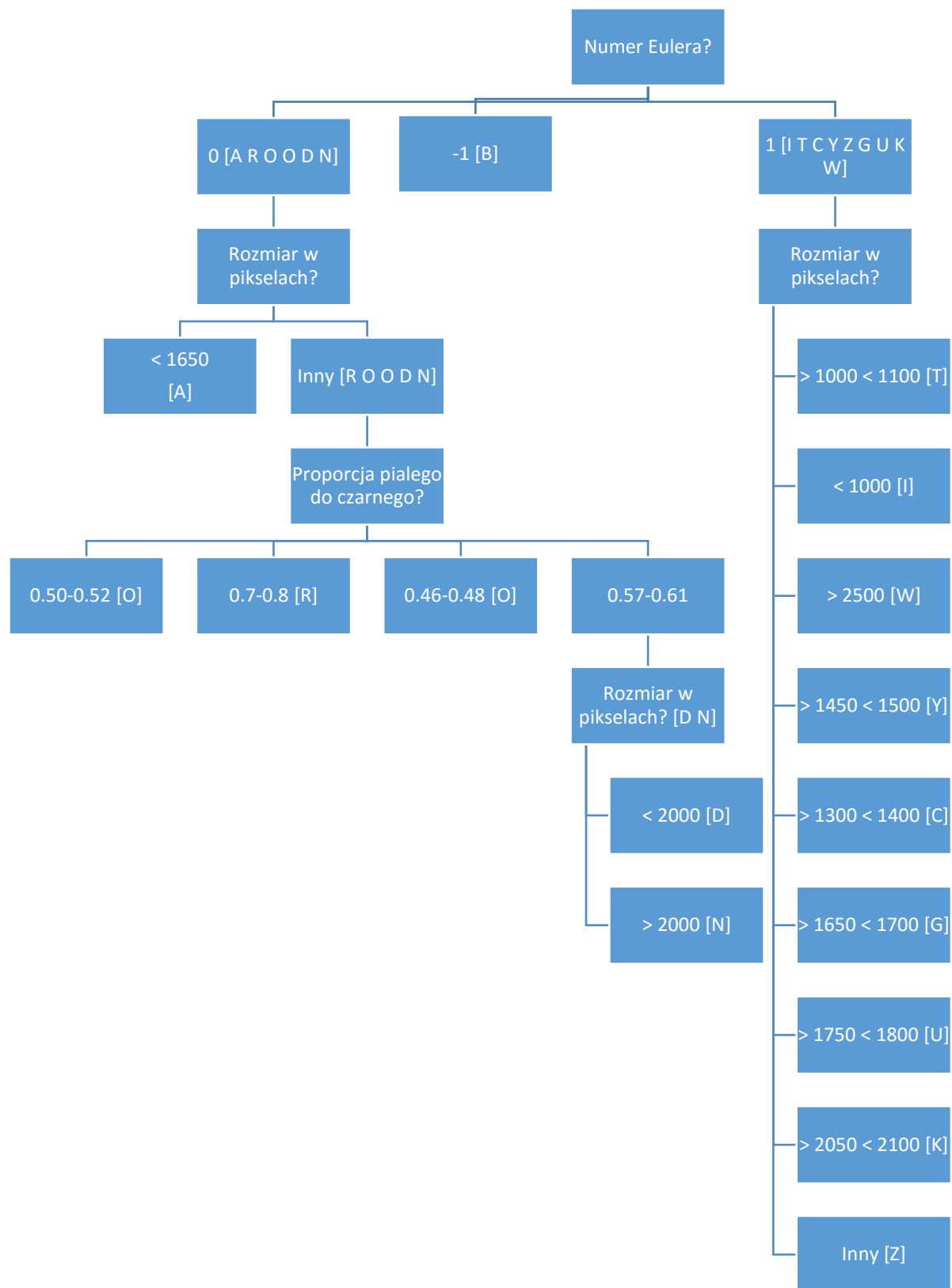


Rys 10. Pokolorowane segmenty

W ten sposób przygotowano obraz do odczytania cech liter.

## 2. Cechy liter

Obliczono numer Eulera na podstawie równania  $V - E = 1$  gdzie V to ilość wierzchołków a E – krawędzi. Wyznaczono pola powierzchni i stosunek białych pikseli do czarnych w kwadracie w którym wpisany jest segment. Litery rozpoznano za pomocą drzewa decyzyjnego przedstawionego poniżej.



Rys 11. Drzewo decyzyjne

Dzięki temu udało się rozpoznać wszystkie litery.

Odczytywanie słów rozpoczęto od odczytania słów pionowych. Następnie odczytano słowa poziome. Ostatecznie odnaleziono słowa w bazie słów które nie występowały na obrazie.

## Dyskusja

Odczytano poprawnie wszystkie słowa. Niestety nie udało się prawidłowo zidentyfikować litery ń dlatego zmieniono słowo „gryzoń” na „gryzon” w bazie słów. Być może udałoby się to przy użyciu bardziej zaawansowanego algorytmu który uwzględniałby segmenty położone blisko siebie jako potencjalnie należące do jednej litery. Poswiecenie litery ń wpłynęło pozytywnie na usunięcie szumu który był prawie całkowicie nieobecny o obrazie końcowym na którym rozpoznawano litery.