Jakub Kwiatkowski

**Rozpoznawanie rodzajów odpadów za pomocą drzewa decyzyjnego**

**Język:** Python

**Biblioteki:** PIL, cv2, graphviz

**Algorytm:** ID3

**Zbiór uczący na podstawie:**

* 100 zdjęć odpadów plastikowych
* 100 zdjęć odpadów szklanych
* 100 zdjęć odpadów papierowych
* Źródło: <https://github.com/garythung/trashnet>

**Atrybuty zbioru uczącego:**

* ilość określonych kolorów (0.0-0.9) po zredukowaniu kolorów (9)
* momenty Hu (przeskalowane i zaokrąglone do liczb całkowitych) po zredukowaniu kolorów (2)

**Decyzje:**

* plastic
* glass
* paper

**Klasy:**

|  |  |  |  |
| --- | --- | --- | --- |
| **GarbageClassifier** | **Example** | **ImageExample** | **Node** |
| **tree**  **examples[]**  **attributes** | attributes  decision | decision  **whitePixels…**  **huMoments**  colors  binaryImage  colorImage | edge  label  children[] |
| getMostCommonClass(e)  checkIfOneClassOnly(e)  countAttributeEntropy(a, e)  getTree()  buildTree(pE, d, pA)  printWholeTree()  printTree(t,s,f)  test(string) |  | getDistance()  countPixels()  getNearestColor(pixelColor)  changeImageColors()  transformImage()  setHuMoments()  **getString()** |  |

**Krótki opis działania:**

1. \*Przetworzenie obrazów ze zbioru uczącego i zapisanie informacji o nich (kolory, momenty Hu) przy pomocy metod klasy ImageExample w pliku tekstowym.
2. Utworzenie instancji klasy GarbageClassifier - załadowanie informacji z pliku tekstowego i zapisanie ich jako zbiór uczący (lista instancji klas Example, w których atrybuty i wartości są reprezentowane w słowniku). Zbudowanie drzewa decyzyjnego przy pomocy algorytmu ID3.
3. Testowanie metodą test, która jako argument przyjmuje tekst powstały po przetworzeniu testowego obrazu i przeszukuje powstałe drzewo (Node)