**Obrázok, na ktorom je umenie

Automaticky generovaný popisŽILINSKÁ UNIVERZITA V ŽILINE**

Fakulta riadenia a informatiky

Softvérové spracovanie biomedicínskych údajov

Semestrálna práca

5ZIB11 Branislav Caban

2023

# Zadanie 1 – vytvorenie pipeline v CellProfiler

Dáta: <https://bbbc.broadinstitute.org/BBBC022>, 50 obrázkov

Pipeline som vyskladal z nasledujúcich modulov:

**EnhanceOrSuppressFeatures –** umožňuje vylepšiť určité funkcie v obrázku, aby sa zlepšila následná analýza obrázkov. Smoothing scale hovorí o množstve použitých filtrov, ktoré pomáhajú odstrániť šumy. Nastavil som tento parameter na 5.0.

**RescaleIntensity –** používa sa na normalizáciu intenzity pixelov na rôznych obrázkoch alebo na zvýšenie kontrastu obráka.

**IdentifyPrimaryObjects –** slúži na identifikovanie a segmentovanie objektov na obrázku. Pri tomto module som postupoval takým spôsobom, že som najprv odmeral objekty na obrázku a podľa toho som nastavil minimálny priemer na 18 pixelov a maximálny na 80 pixelov. Dôležité bolo nastaviť hlavne ten minimálny, pretože ak to bolo moc nízke číslo, spozoroval som, že niektoré objekty bralo ako 2 rôzne aj keď bol to iba 1 objekt. Ďalej som upravil hornú hranicu tresholdu na 0.1, pretože 1.0 bolo zbytočne veľa podľa toho čo som skúšal. A metódu na odlišovanie zhlukovaných objektov som použil „Shape“, vďaka čomu som docielil to, že objekty, ktoré boli spojené alebo boli veľmi blízko seba, označilo správne ako 2 rôzne objekty.

Obrázok, na ktorom je text, snímka obrazovky, pestrofarebnosť

Automaticky generovaný popis

**OverlayOutlines –** prekrýva obrysy identifikovaných objektov na pôvodný obrázok. Tento modul sa často používa na účely vizualizácie na vizuálne zvýraznenie a overenie presnosti identifikácie objektov a výsledkov segmentácie.

**MeasureObjectIntensity** – používa sa na kvantitatívne meranie intenzity objektov v obraze. Napr. stredná intenzita, celková intenzita, minimálna / maximálna intenzita a iné.

Obrázok, na ktorom je text, snímka obrazovky, číslo, jedálny lístok

Automaticky generovaný popiss

**ExportToSpreadsheet –** uloženie výsledkov do tabuľkového formátu.

# Zadanie 2 – R script

V tomto scripte som spravil logistickú regresiu na dátach o cukrovke. Obsahuje 8 nezávislých premenných a 1 závislú premennú. Nezávislé premenné: **Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age.** Závislá premenná: **Outcome –** teda výsledok či dostal cukrovku alebo nie (0 alebo 1).

Použil som 100 dát, ktoré som si rozdelil na trénovacie a testovacie v pomere 8:2. Presnosť na testovacích dátach vyšla 0.7, ktorú som získal po použití metódy predict() a následnom použití tresholdu - ak výsledok je väčší ako 0.5 tak je to 1 inak 0. Treshold je potrebný, keďže chceme mať binárny výstup. Na nasledujúcom histograme vidíme, ktoré nezávisle premenné majú najväčšiu rolu pri predikovaní cukrovky.

Obrázok, na ktorom je text, snímka obrazovky, diagram, vývoj

Automaticky generovaný popis

# Zadanie 3 – Algoritmus v Pythone

Pre túto časť som si vybral algoritmus, ktorý bude segmentovať obrázky tkanív prsníka a následne vyhodnotí DICE skóre ako veľmi presná bola segmentácia. Keďže vychádzam z problematiky, ktorú riešime na projekte 2, k dispozícii mám priečinok ground-truth\_images, kde sú obrázky ako by to malo byť. Tieto obrázky sú rozdelené do 3 skupín: tumorová oblasť, stromová oblasť a zvyšok. Taktiež mám k dispozícii json súbor, kde je ku každému obrázku povedaný počet týchto oblastí. Toto je dôležité, pretože aby som mohol aplikovať k-means algoritmus potrebujem vedieť na koľko zhlukov to mám segmentovať a každý obrázok obsahuje iný počet oblasti. Použil som 97 dát.

Ako prvé som vytvoril algoritmus, ktorý prečíta spomínaný json súbor a zistí ku každému obrázku koľko oblastí sa tam nachádza a následne túto informáciu použije pri k-means algoritme, kde je povinný parameter počet zhlukov. Výsledné obrázky uložím do priečinka clustered\_images.

V priečinku ground-truth\_images mám obrázky v 3 farbách: modrá pre stromovú oblasť, zelená pre nádorovú oblasť a červená pre zvyšok (čo ma nezaujíma). Takže ďalší krok je prefarbenie výsledných obrázkov do týchto farieb. Vypozoroval som, že najsvetlejšia farba patrí do stromálnej oblasti, menej svetlá farba do nádorovej oblasti a tá najtmavšia pre zvyšok. Vytvoril som si json súbor, kde som si zoradil RGB týchto farieb v poradí od najsvetlejšej farby po najtmavšiu. Následne som z tohto json súboru čítal tieto farby a postupne som ich zafarboval, tak aby sa to dalo porovnávať s ground-truth\_images. Teda najsvetlejšiu farbu som zafarbil na modrú, menej svetlu na zeleno a najtmavšiu na červeno.

Ostáva už len výpočet DICE skóre, čo je metrika, na vyhodnotenie presnosti segmentácie, kde vstupuje výsledný obrázok a správny obrázok. Počíta sa osobitne DICE pre nádorovú a stromálnu oblasť a následne sa spraví priemer a máme konečné DICE skóre. Výsledok mi vyšiel 49%.

Postupnosť krokov pre spustenie:

1. Treba zmazať priečinok clustered\_images a json súbor sorted\_roi-level-cells ak existujú
2. Spustiť k-means.py
3. Spustiť clustered\_colors.py
4. Spustiť dice-score.py

Ukážka výpočtu dice score pre stromovú oblasť.

dice\_score\_stroma = 2 \* match\_stroma / all\_counter\_stroma if all\_counter\_stroma > 0 else 0

Použitie k-means z OpenCV knižnice.

ret, label, center = cv.kmeans(Z, K, None, criteria, 10, cv.KMEANS\_RANDOM\_CENTERS)