Specyfikacja implementacyjna

Program do zliczania komórek na zdjęciach mikroskopowych

Paweł Skiba, 290997

29 marca 2021

Spis treści

1 Opis ogólny **2**
- 1.1 Koncepcja budowy programu . . . 2
- 1.2 Technologie . . . 2
- 1.3 Konwencja nazw . . . 2

2 Diagram **3**
- 2.1 Diagram modułów . . . 3

3 Pakiety i moduły **3**
- 3.1 Pakiety . . . 3
- 3.2 Podział modułów . . . 3
- 3.3 Moduły cyfrowego przetwarzania obrazów . . . 4
- 3.4 Moduły sieci neuronowej . . . 4
- 3.5 Moduły współdzielone . . . 4

4 Algorytm **5**
- 4.1 Cyfrowe przetwarzanie obrazów . . . 5
- 4.2 Sieć neuronowa . . . 6

5 Testowanie **7**
- 5.1 Proces testowania . . . 7
- 5.2 Cyfrowe przetwarzanie obrazów . . . 7
- 5.3 Sieć neuronowa . . . 7

1 Opis ogólny

1.1 Koncepcja budowy programu

Projekt jest związany ze stworzeniem optymalnego algorytmu do zliczania komórek na zdjęciach mikroskopowych, przez co trudno jest określić jego dokładną implementację. Z tego powodu, możliwe będzie stworzenie ogólnego zarysu programu oraz jego modułów. Finalnie program oraz zaproponowane podejście do zaprojektowania algorytmu może ulec zmianie.

1.2 Technologie

Program główny zostanie zaimplementowany w języku *Python* w sposób proceduralny, bądź imperatywny. W przypadku cyfrowego przetwarzania obrazów, algorytm zostanie zrealizowany również w języku *Python*, natomiast istnieje możliwość wykorzystania w pewnych obszarach jego obiektowości. W przypadku sieci neuronowych również zostanie wybrana implementacja w języku *Python*. Opracowanie obydwu algorytmów zostanie wstępnie zaimplementowane w środowisku *Jupyter Notebook*. Program będzie można uruchomić z poziomu wiersza poleceń.

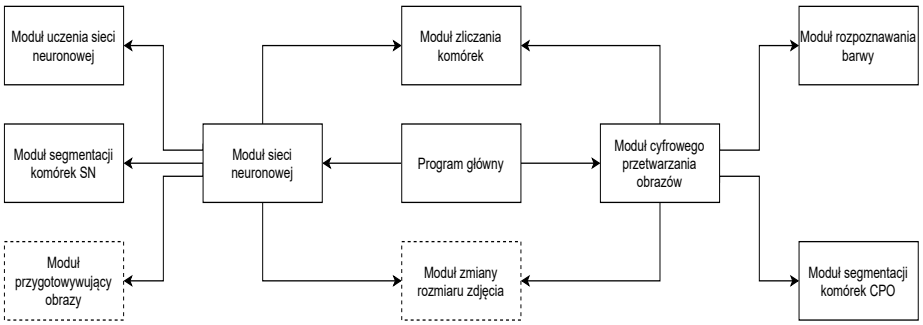
1.3 Konwencja nazw

Program zostanie zaimplementowany w języku angielskim, co pozwoli na ujednolicenie kodu programu. Dzięki temu kod programu będzie bardziej uniwersalny oraz będzie zrozumiały dla większej liczby osób. Ponadto, uważane jest to za dobrą praktykę.

2 Diagram

2.1 Diagram modułów

Poniżej zostanie zaprezentowany diagram modułów, które zostaną zaimplementowane w programie. Moduły oznaczone linią przerywaną oznaczają moduły, które mogą zostać zaimplementowane, lecz nie ma pewności co do ich niezbędności.



Rysunek 1: Diagram modułów

3 Pakiety i moduły

3.1 Pakiety

W celu uporządkowania projektu powstaną cztery pakiety, w których będą przechowywane pliki.

- main – pakiet będzie zawierał program główny,
- dip – pakiet będzie zawierał moduły/pliki związane z cyfrowym przetwarzaniem sygnałów,
- nn – pakiet będzie zawierał moduły/pliki związane z sieciami neuronowymi,
- shared – pakiet będzie zawierał moduły/pliki, które będą wspólne dla cyfrowego przetwarzania sygnałów i sieci neuronowych.

3.2 Podział modułów

Moduły możemy podzielić na trzy następujące kategorie – moduły dla cyfrowego przetwarzania obrazów, moduły dla sieci neuronowych oraz moduły wspólne dla dwóch różnych podejść. Każdy z modułów będzie osobnym plikiem, który będzie miał określone zadanie do wykonania oraz jedną lub więcej zaimplementowanych funkcji. Każdy moduł będzie zawarty w odpowiednim pakiecie. Poniżej zostanie opisana koncepcja każdego z modułów.

3.3 Moduły cyfrowego przetwarzania obrazów

- Moduł cyfrowego przetwarzania obrazów
  Nazwa modułu: *digital_image_processing.py*
  Moduł jest odpowiedzialny za uruchamianie innych modułów w obszarze cyfrowego przetwarzania sygnałów.

- Moduł rozpoznawania barwy
  Nazwa modułu: *colour_recognition.py*
  Moduł jest odpowiedzialny za poprawne rozpoznanie, czy obraz składa się z pikseli w odcieniach szarości, czy z pikseli kolorowych.

- Moduł segmentacji komórek CPO
  Nazwa modułu: *segmentation_dip.py*
  Moduł jest odpowiedzialny za przygotowanie zdjęcia z wysegmentowanymi komórkami gotowymi do zliczenia.

3.4 Moduły sieci neuronowej

- Moduł sieci neuronowej
  Nazwa modułu: *neural_network.py*
  Moduł jest odpowiedzialny za uruchamianie innych modułów w obszarze sieci neuronowych.

- Moduł uczenia sieci neuronowej
  Nazwa modułu: uzależniona od implementacji sieci neuronowej
  Moduł ten zostanie dostarczony przez wybraną implementację sieci neuronowej oraz będzie służył do uczenia sieci neuronowej.

- Moduł segmentacji komórek SN
  Nazwa modułu: uzależniona od implementacji sieci neuronowej
  Moduł ten zostanie dostarczony przez wybraną implementację sieci oraz będzie służył do segmentacji komórek na zdjęciach mikroskopowych.

- *Moduł przygotowywujący obrazy
  Nazwa modułu: *prepare_images.py*
  W przypadku potrzeby powstania tego modułu, będzie on odpowiedzialny za przygtowania danych do uczenia sieci neuronowj, to znaczy dostosowania aktualnych katalogów z obrazami i maskami do wybranej implementacji sieci neuronowej.

3.5 Moduły współdzielone

- Moduł zliczania komórek
  Nazwa modułu: *counter.py*

Moduł jest odpowiedzialny za zliczenie wysegmentowanych komórek, zarówno w przypadku cyfrowego przetwarzania obrazów, jak i sztucznych sieci neuronowych. Moduł będzie wykorzystywał biblioteki, takie jak *OpenCV*, czy *scikit-image*.

- *Moduł zmiany rozmiaru zdjęć
  Nazwa modułu: *change_images_size.py*
  W przypadku potrzeby powstania tego modułu, będzie on odpowiedzialny za zmianę rozmiaru zdjęć, do takiej przy której będą uzyskiwane najlepsze wyniki. Moduł może zostać wykorzystany w przypadku potrzeby normalizacji rozmiaru zdjęć dla sieci neuronowej.

4 Algorytm

Algorytmy każdego z dwóch różnych podejść do rozwiązania problemu będą przede wszystkim zaimplementowane w modułach segmentacji komórek CPO i SN. W przypadku cyfrowego przetwarzania obrazów, dodatkowo możemy włączyć w to zagadnienie moduł rozpoznawania barw. Poniżej zostaną opisane szczegółowo elementy, które będą wykorzystane w algorytmach.

4.1 Cyfrowe przetwarzanie obrazów

Realizacja algorytmu za pomocą cyfrowego przetwarzania obrazów obejmować będzie dwa moduły, a dokładnie moduł segmentacji komórek CPO oraz moduł rozpoznawania barw, co jest związane z jakością obrazu. W przypadku obrazów w odcieniach szarości, tło jest jednolite (zazwyczaj czarny kolor), dzięki czemu obraz jest dobrej jakości pomimo rozmycia komórek. W przypadku obrazów kolorowych, tło jest bardzo rozmyte, co może wymagać innej obróbki obrazu w celu segmentacji komórek. Z tego powodu, na początku zostanie przeprowadzona kwalifikacja obrazu w celu określenia, czy obraz jest w odcieniach szarości, czy jest kolorowy.

Specyfika projektu nie pozwala na zaprojektowanie algorytmu, ze względu na fakt, iż będzie on tworzony emiprycznie. Natomiast możemy wyróżnić kilka dużych etapów w algorytmie, które zostaną zrealizowane.

1. Usunięcie rozmycia obrazów za pomocą wybranych operacji.
2. Wykrycie krawędzi za pomocą określonych operacji.
3. Segmentacja komórek na obrazie.

W celu zrealizowania powyższych kroków zostaną wykorzystane techniki, takie jak:

- filtracja – pozwala na przetwarzanie obrazu w celu zmiany, bądź uwypuklenia jego cech. Dzięki filtracji możemy usunąć szumy róznego rodzaju oraz poprawić jakość obrazu.
- segmentacja – pozwala na wyodrębnienie części obrazu, która reprezentuje pewien obiekt poprzez zmianę wartości pikseli na taką samą w ramach określonego obiektu. Najprostszym przykładem segmentacji obrazu w odcieniach szarości może być binaryzacja.
- binaryzacja – pozwala na przekształcenie obrazu w jednowymiarowy obraz, gdzie każdy punkt może przyjąć kolor biały bądź czarny. Operacja ta może być przydatna podczas segmentacji. Wykorzystana może zostać metoda Otsu.
- morfologia matematyczna – dziedzina ta pozwala wykonywać operacje na obrazach z wykorzystaniem masek, które mają wpływ na ostateczną wartość danego punktu. Może być wykorzsytana do wykrywania krawędzi za pomocą gradientu morfologicznego.

Projekt algorytmu w znacznej części zostanie oparty o następujące biblioteki:

- numpy – biblioteka wspomagająca obsługę wektorów oraz macierzy,
- matplotlib – biblioteka do tworzenia wykresów,
- skimage (scikit-image) – biblioteka dedykowana do przetwarzania obrazów,
- cv2 (OpenCV) – biblioteka wspierająca przetwarzanie obrazów.

4.2 Sieć neuronowa

Sieci neuronowe są bardzo obszernym tematem. Cechują się one różnorodnością pod względem nie tylko rodzaju sieci neuronowej, ale także implementacji. Określona sieć neuronowa może zostać zaimplementowana w różnych językach, bazując na różnorodnych bibliotekach. W przypadku języka *Python*, do najpopularniejszych możemy zaliczyć biblioteki, takie jak *TensorFlow*, *Keras* czy *PyTorch*.

Na początku należy określić rodzaj sieci neuronowej, jaka zostanie wykorzystana w projekcie w celu segmentacji komórek na zdjęciach mikroskopowych. Przypuszczalnie najlepszym rozwiązaniem będzie wykorzystanie konwolucyjnych sieci neuronowych. Sieci te są wzbogacone o dodatkowe możliwości ekstrakcji, bądź uwypuklenia cech na obrazach. Jest to efekt określonych filtracji sygnałów zastosowanych na obrazach. Możemy osiągnąć taki efekt, ze względu na fakt, iż cyfrowe przetwarzanie obrazów jest częścią cyfrowego przetwarzania sygnałów, natomiast przetwarzane sygnały możemy traktować jako obraz.

Kolejnym etapem jest wybór konkretnej konwolucyjnej sieci neuronowej. W tym celu zostaną zaproponowane dwa rozwiązania, a ostateczna decyzja zostanie podjęta na podstawie wstępnego zapoznania się z określonymi sieciami neuronowymi. Poniżej zostaną krótko opisane wybrane wstępnie sieci neuronowe.

- U-Net – konwolucyjna sieć neuronowa, zaprojektowana do segmentacji obrazów biomedycznych na Uniwersytecie we Freiburgu.
- Mask R-CNN (Mask Region Based Convolutional Neural Networks) – konwolucyjna sieć neuronowa pochodząca z rodziny sieci neuronowych R-CNN, która przestała koncentrować się na wykrywaniu obiektów, na rzecz ich segmentacji. Jest ona rozwijana przez firmę *Facebook*.

Ostatnią rzeczą jaką należy okrelić w przypadku sieci neuronowej, to jej implementacja. Niezależnie od wyboru sieci neuronowj (pierwsza, bądź druga opisana sieć neuronowa) zostanie wybrana implementacja z użyciem biblioteki *PyTorch*.

5 Testowanie

5.1 Proces testowania

Proces testowania jest całkowicie uzależniony od podejścia do rozwiązania problemu. Dlatego poniżej zostaną zaprezentowane dwie różne propozycje przeprowadzenia procesu testowania.

5.2 Cyfrowe przetwarzanie obrazów

W przypadku cyfrowego przetwarzania obrazów, możemy wykorzystać fakt, iż w zbiorze jest zestaw *stage1_train* przeznaczony do trenowania sieci neuronowej. Zestaw ten oprócz obrazów z komórkami zawiera zestaw masek dla każdego z obrazów, co pozwala nam zidentyfikować liczbę komórek na zdjęciu mikroskopowym. Rodzaj tej metody pozwala nam wykorzystać uprzednio wymieniony fakt, ze względu na to, że nie wymaga ona procesu uczenia, a zatem oryginalne zdjęcia oraz maski nie zostaną wcześniej przez nią wykorzystane.

Niewykluczone, że w celu zautomatyzowania tego zadania powstanie skrypt w języku *Python* o nazwie *count_accuracy.py*, który będzie w stanie wyznaczyć dokładność algorytmu. Skrypt ten będzie wyznaczał dokładność dla jednego obrazu, jako stosunek komórek znalezionych przez program do liczby masek dla danego obrazu. Następnie wynik zostanie uśredniony dla określonej próbki obrazów.

5.3 Sieć neuronowa

W przypadku sieci neuronowej możemy zastosować dwa podejścia. Pierwsze z nich wymaga zidentyfikowania oraz manualnego zliczenia komórek na obra-

zach ze zbioru *stage1_test* oraz porównania ich z otrzymanymi wynikami.

Dodatkowo można wyznaczyć część zbioru, która będzia określona jako podzbiór walidacyjny. Bardzo często takie rozwiązanie jest stosowane, natomiast jest ono uzależnione od wybranej implementacji sieci neuronowej. W takiej sytuacji, użytkownik określa jaki procent obrazów posłuży jako zbiór walidacyjny, a następnie sieć neuronowa wyznacza sama dokładność na podstawie wysegmentowanych elementów.