Implementacja równoległego mechanizmu wyliczania indeksów wilgotności (NDMI) i wegetacji (NDVI) roślin na zobrazowaniach satelitarnych

Sprawozdanie z projektu

Spis treści

Wstęp

Celem projektu jest stworzenie programu wyposażonego w GUI, umożliwiającego równoległe wyliczanie wskaźników: NDVI i NDMI. Wskaźniki te obliczane są na podstawie danych z satelity Sentinel-2.

NDVI, czyli znormalizowany różnicowy wskaźnik wegetacji, to miara kondycji i gęstości roślinności. Bazuje na kontraście między silnym pochłanianiem promieniowania czerwonego (Red) przez chlorofil, a silnym odbiciem promieniowania bliskiej podczerwieni przez struktury komórkowe liści.

NDMI, czyli znormalizowany różnicowy wskaźnik wilgotności, to miara zawartości wody w roślinności. Bazuje na kontraście między silnym odbiciem promieniowania bliskiej podczerwieni (NIR) przez wewnętrzną strukturę liści, a silnym pochłanianiem promieniowania krótkofalowej podczerwieni (SWIR) przez wodę zawartą w komórkach roślinnych.

NDVI i NDMI

Do obliczenia wskaźników użyto następujących wzorów:

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Gdzie

- NIR – odbicie w paśmie bliskiej podczerwieni (Near Infrared)
- Red – odbicie w paśmie czerwonym światła widzialnego

$$NDMI = \frac{NIR - SWIR}{NIR + SWIR}$$

Gdzie

- NIR – odbicie w paśmie bliskiej podczerwieni (Near Infrared)
- $SWIR$ – odbicie w paśmie krótkofalowej podczerwieni (Shortwave Infrared)

Oba wskaźniki mają wartości należące do przedziału $[-1, 1]$, wartości ich w programie są interpolowane do skali kolorów.

Dane

Dane wejściowe do analizy pochodzą z misji Sentinel-2 i zostały pobrane za pośrednictwem platformy: Copernicus Browser. Aplikacja działa zarówno z danymi L1C, jak i L2A, które są

dostępne na stronie Copernicus Browser. Do rozpoczęcia obliczeń wskaźników należy podać 4 pasma:

- B04 – pasmo RED (rozdzielczość 10m)
- B08 – pasmo NIR (rozdzielczość 10m)
- B11 – pasmo SWIR (rozdzielczość 20m)
- SCL – warstwa klasyfikacji sceny (rozdzielczość 20m)

SCL nie jest typowym pasmem spektralnym, ale mapą klasyfikacyjną. Każdemu pikselowi obrazu przypisuje jedną z predefiniowanych kategorii. Dzięki temu można wykluczyć obszary, które są zasłonięte chmurami, wodę lub śnieg. Poniżej znajduje się opis informujący, które wartości maski zostały wykluczone. Wykluczone piksele będą zaznaczone na mapie wynikowej kolorem czarnym.

Wykluczane wartości:

- 0: no data
- 1: saturated defective
- 3: cloud shadow
- 6: water
- 8: cloud medium probability
- 9: cloud high probability
- 10: thin cirrus
- 11: snow ice

Niewykluczane wartości:

- 2: topographic casted shadows
- 4: vegetation
- 5: not vegetated
- 7: unclassified

Oprogramowanie

Program został napisany w języku C. Do operacji na danych geoprzestrzennych, takich jak wczytywanie obrazów satelitarnych Sentinel-2 (pliki .jp2), użyto biblioteki GDAL. Interfejs graficzny użytkownika został zaimplementowany za pomocą biblioteki GTK. W celu przyspieszenia obliczeń, zaimplementowano wielowątkowe przetwarzanie równoległe przy użyciu biblioteki OpenMP. Ta forma zrównoleglenia została wybrana ze względu na

niezależność danych, powstałą po dekompozycji funkcjonalnej, oraz prostotę implementacji omp.

Uruchomienie

Program uruchamia się komendami:

```
$ make                  # Kompilacja
$ ./program.out         # Uruchomienie
```

Kroki wykonania programu

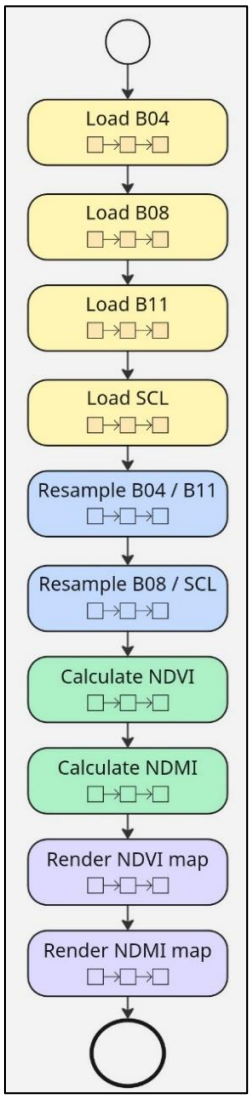


Diagram 1 Kroki wykonania programu sekwencyjnego

Na powyższym diagramie przedstawiono kroki sekwencyjnego wykonania programu. Składa się on z:

1. Wczytania plików zawierających odpowiednie pasma zdjęć satelitarnych (B04, B08, B11, SCL),
2. Przepróbkowania pasm do jednakowej rozdzielczości (upsampling 10m lub downsampling 20m),
3. Obliczenia wskaźników NDVI oraz NDMI,
4. Wygenerowania wynikowych map wskaźników.

Rysunki wewnątrz prostokątów oznaczają, że dana operacja wykonuje się **sekwencyjnie**.

Dla operacji resamplingu zaimplementowano następujące algorytmy:

- B04, B08 – downsampling 10m→20m – interpolacja uśredniająca
- B11 – upsampling 20m→10m – interpolacja liniowa
- SCL – upsampling 20m→10m – interpolacja metodą najbliższego sąsiada (KNN dla k = 1)

Moduły implementujące program

- `data_loader` – wczytywanie pasm z plików `.jp2`
- `data_saver` – zapisywanie map gdk do plików `.png`
- `resampler` – przepróbkowanie obrazu do zadanej rozdzielczości
- `index_calculator` – obliczenie NDVI oraz NDMI
- `visualization` – obliczanie kolorów pikseli mapy wskaźników
- `processing_pipeline` – wykonanie kroków programu
- `gui` – interfejs graficzny

Zrównoleglenie procesów

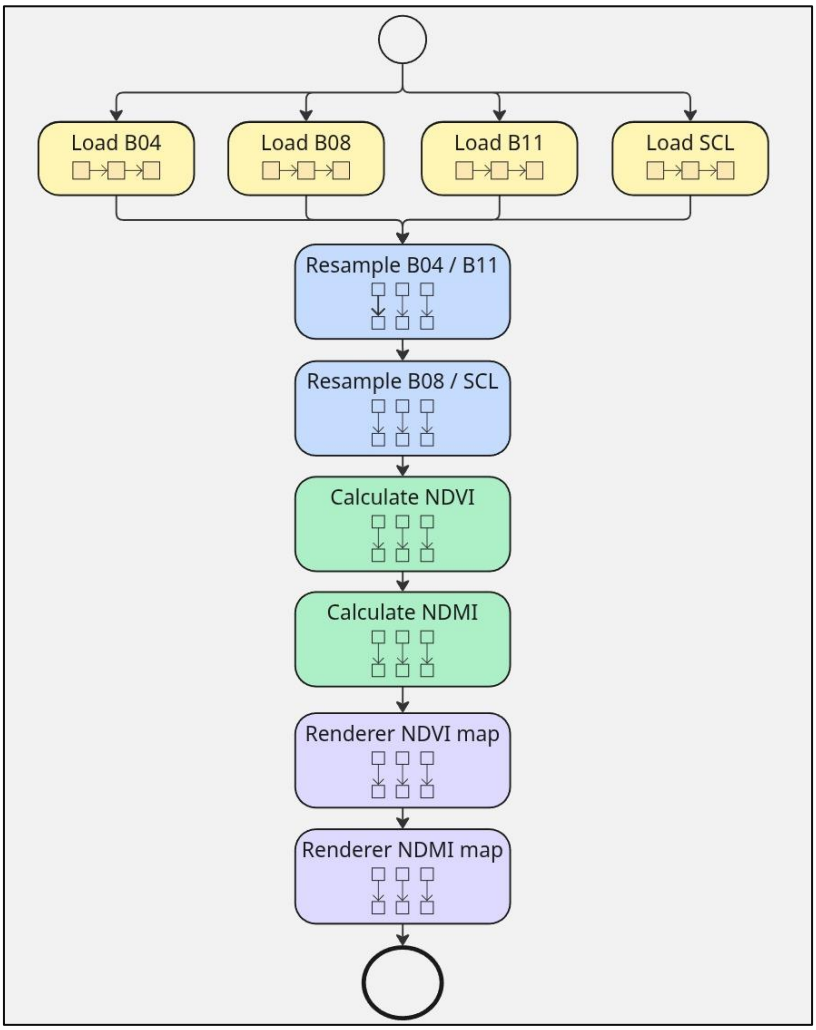


Diagram 2 Kroki wykonania programu równoległego

Rysunki w prostokątach na powyższym diagramie oznaczają, że przetwarzanie danych w danej operacji odbywa się **sekwencyjnie** (wczytywanie) lub **równolegle** (obliczenia).

Wprowadzone zmiany:

- Wczytywanie wszystkich 4 plików ze zdjęciami satelitarnymi wykonuje się równolegle, gdzie każde pasmo jest obsługiwane przez osobny wątek.
- Resampling pasm jest zrównoleglony poprzez podział danych między wątki, z których każdy przetwarza przypisany mu fragment.

- Obliczanie danego współczynnika NDVI lub NDMI wykonuje się równolegle: dane wejściowe są dzielone, a każdy wątek oblicza wartości wskaźnika dla przydzielonej mu części obrazu.
- Wygenerowanie wynikowej mapy wskaźnika wykonuje się równolegle, również poprzez podzielenie danych na wątki.

Nie zrównoleglono resamplingu 2 pasm jednocześnie, ponieważ zrównoleglenie jednego resamplingu i tak jest zadaniem, które pochłonie wszystkie dostępne rdzenie procesora. Rozłożenie wątków po równo na 2 zadania da taki sam efekt, jak przypisanie wszystkich wątków do jednego zadania i wykonanie tych dwóch zadań po kolei. Ta sama zasada tyczy się pozostałych kroków programu (obliczania współczynników i generowania map).

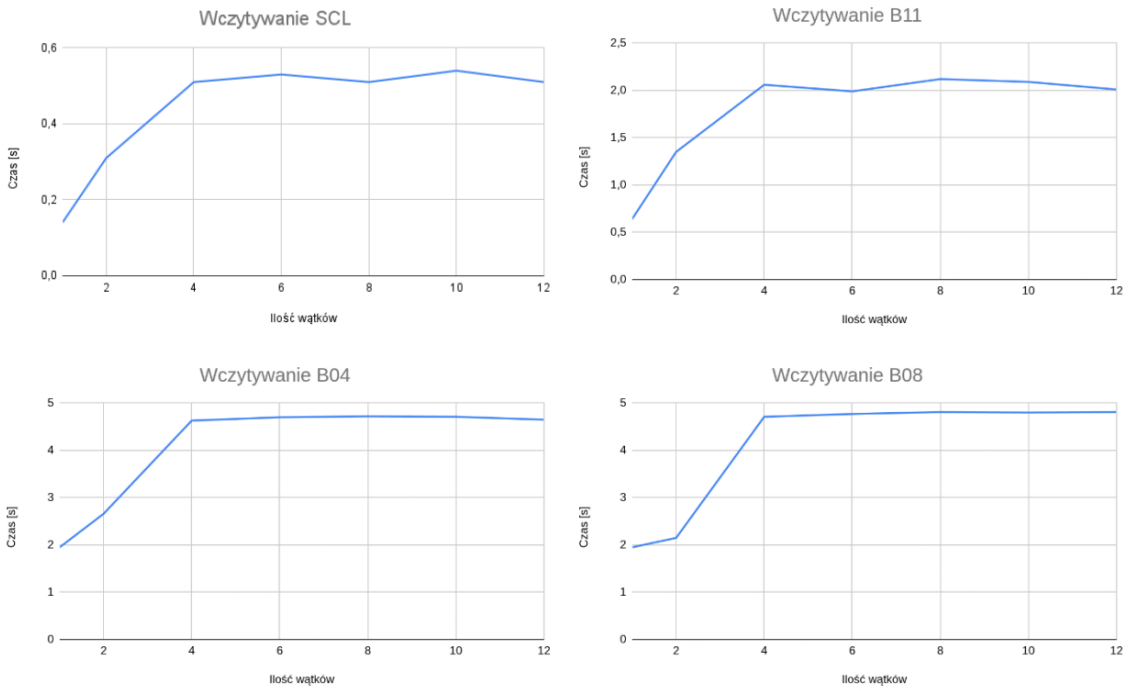
Do zrównoleglenia programu i podziału danych między wątki wykorzystano bibliotekę `openMP`, zwłaszcza dyrektywę

```
#pragma omp parallel for
```

Analiza przyspieszenia

Wykonano pomiary czasowe dla każdego etapu programu. Uzyskano po 3 pomiary dla upsamplingu i downsamplingu dla liczby wątków {1, 2, 4, 8, 10, 12}. Wykresy przedstawiają uśrednione wyniki.

Wczytywanie pasm



Rysunek 1 Zestawienie czasu wykonania wczytywania pasm

Rysunek 2 Całkowity czas wczytywania pasm

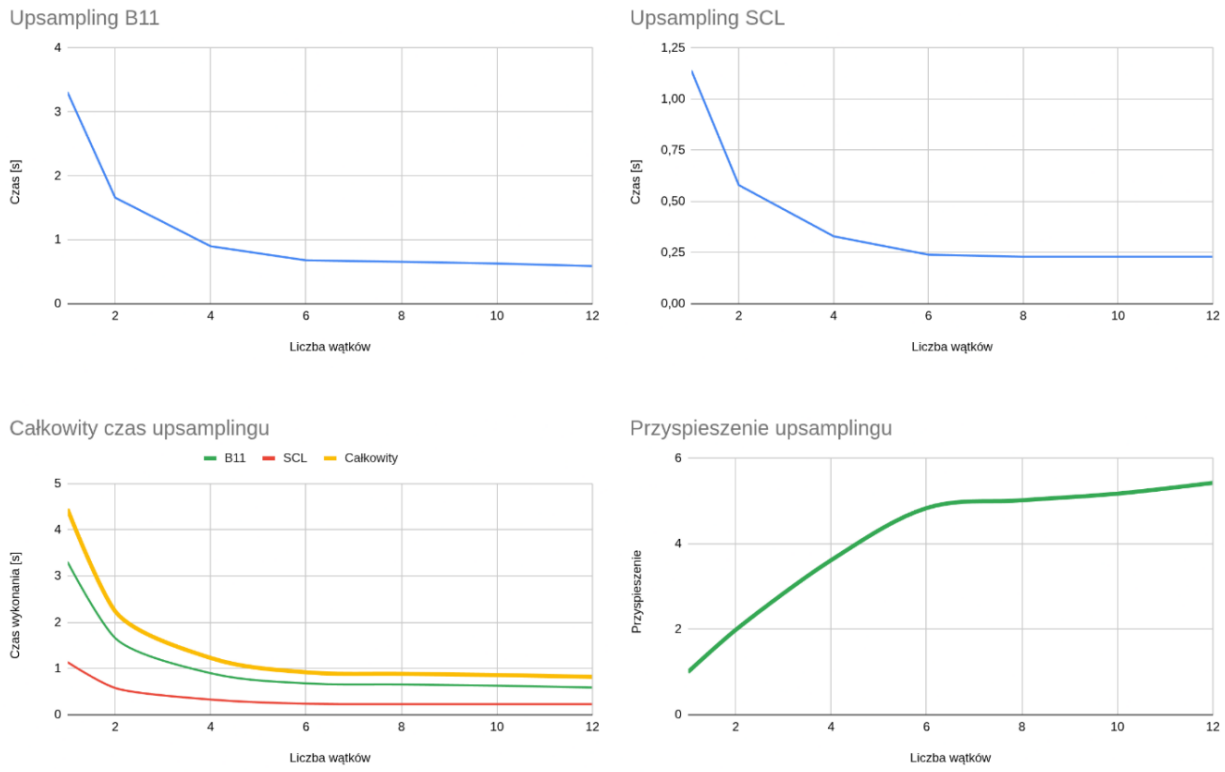
Pomimo równoległego wczytywania wszystkich pasm jednocześnie, **całkowity czas pozostał bez zmian**. Prawdopodobnie wynika to z ograniczeń hardware'owych.

Z wykresów wynika, że czas pierwszego wczytywanego pasma **SCL** jest bez większych zmian, ale czas kolejnego pasma **B11** zaczyna znacząco rosnąć. Dla kolejnych pasm analogicznie.

Na podstawie wyników można wysunąć wnioski, że dysk pomimo obsługi ładowania 2 plików na raz, i tak najpierw czeka na pełne wczytanie poprzedniego pliku. Argumentem za tą tezą jest to, że całkowity czas się nie zmienił. Nie ważne, czy wczytujemy pliki 1 wątkiem czy 12 wątkami, dysk i tak nie obsłuży tej operacji szybciej. Warto zaznaczyć, że czas wczytywania poszczególnych pasm jest stały dla liczby wątków większej niż 4, ponieważ są maksymalnie 4 zadania (4 pasma do wczytania), więc reszta wątków jest bezczynna.

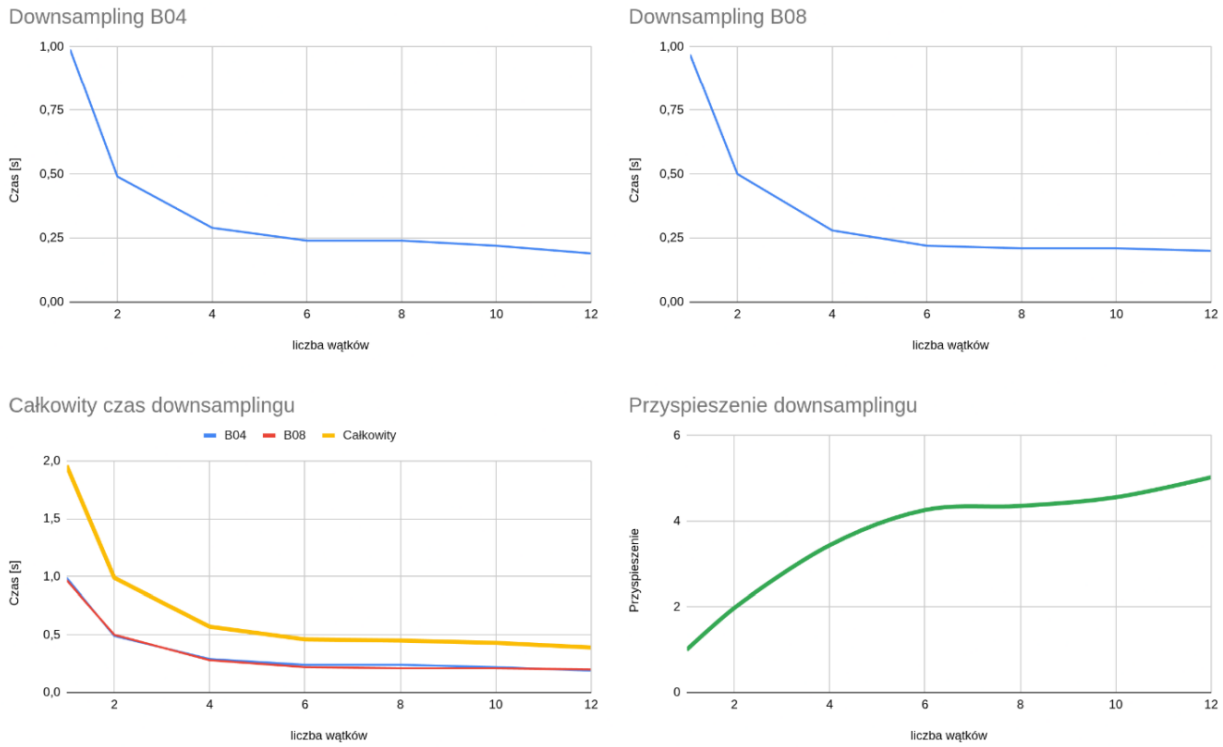
To ograniczenie można by obejść korzystając z 4 różnych dysków lub 4 komputerów, również z osobnymi dyskami.

Resampling



Rysunek 3 Zestawienie czasu wykonania oraz przyspieszenie upsamplingu

Uzyskaliśmy przyspieszenie równe 542,68% przy użyciu 12 wątków. Krzywa przyspieszenia zaczyna się wypłaszczać na poziomie 6-8 wątków.

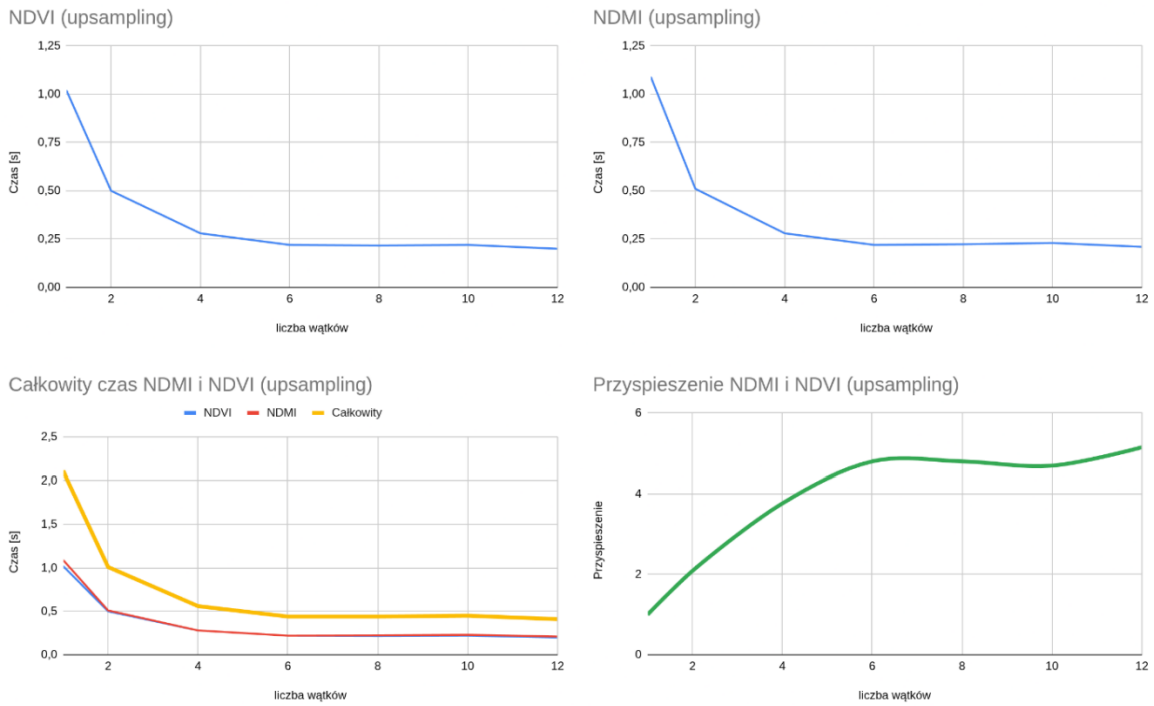


Rysunek 4 Zestawienie czasu wykonania oraz przyspieszenie downsamplingu

Uzyskaliśmy przyspieszenie równe 502,56% przy użyciu 12 wątków. Wypłaszczenie przyspieszenia jest jeszcze bardziej gwałtowne (poza wynikiem dla 12 wątków).

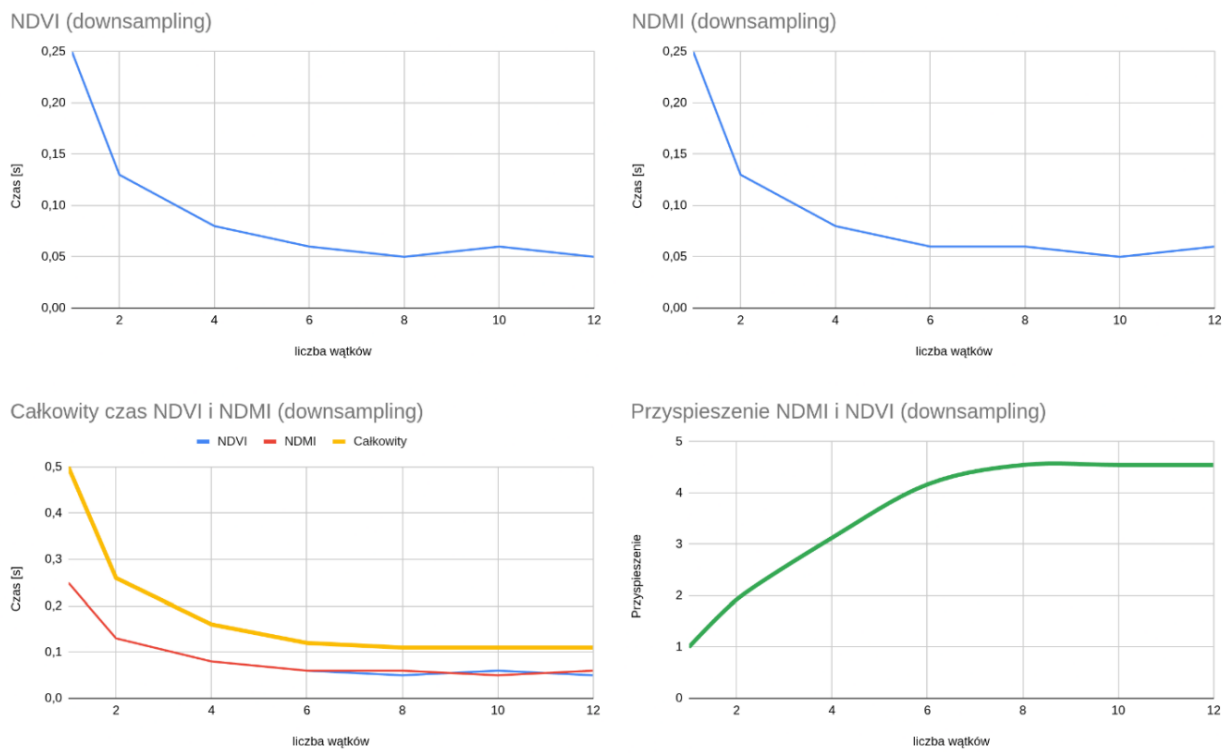
Maksymalne przyspieszenie jest niższe w porównaniu do upsamplingu, ponieważ downsampling wykonuje dużo mniej operacji i narzut tworzenią wątków staje się bardziej znaczący. Różnice pomiędzy uzyskanymi wynikami powyżej 6 wątków są na poziomie 10ms.

Obliczanie wskaźników



Rysunek 5 Zestawienie czasu wykonania oraz przyspieszenie obliczania wskaźników (upsampling)

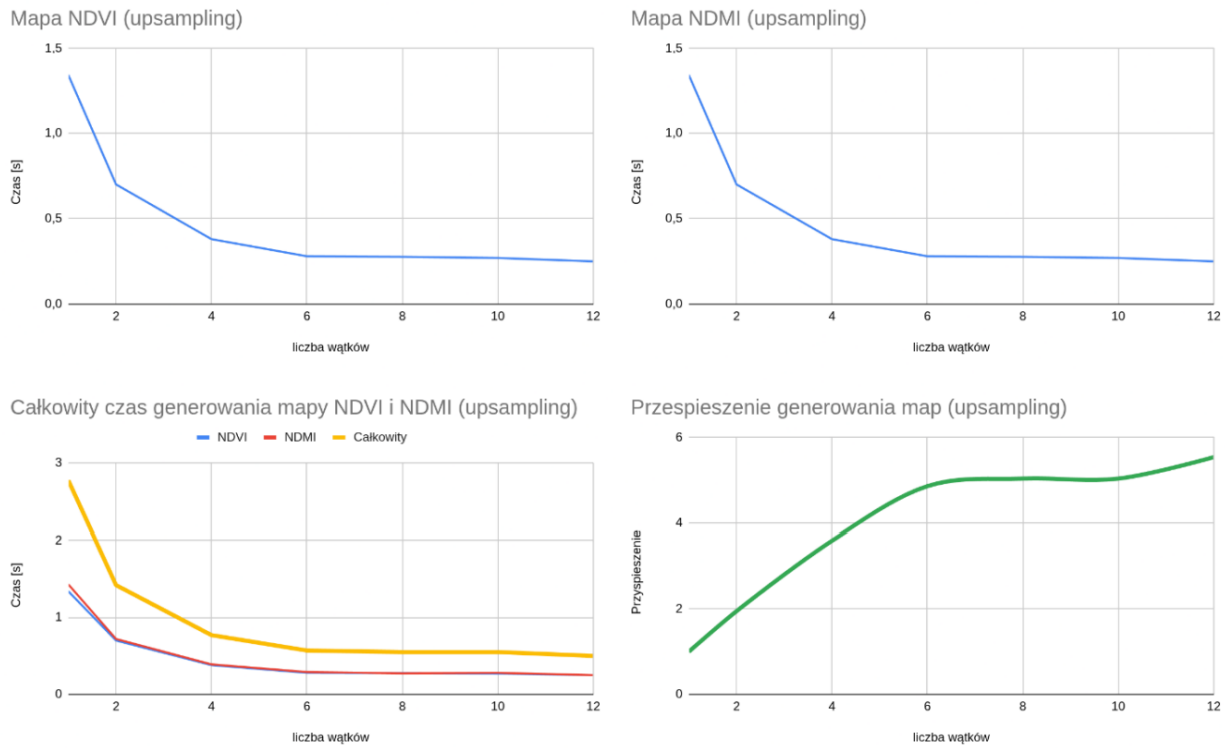
Uzyskaliśmy przyspieszenie równe 514,63% przy użyciu 12 wątków. Przyspieszenie powyżej 6 wątków nie jest znaczne.



Rysunek 6 Zestawienie czasu wykonania oraz przyspieszenie obliczania wskaźników (downsampling)

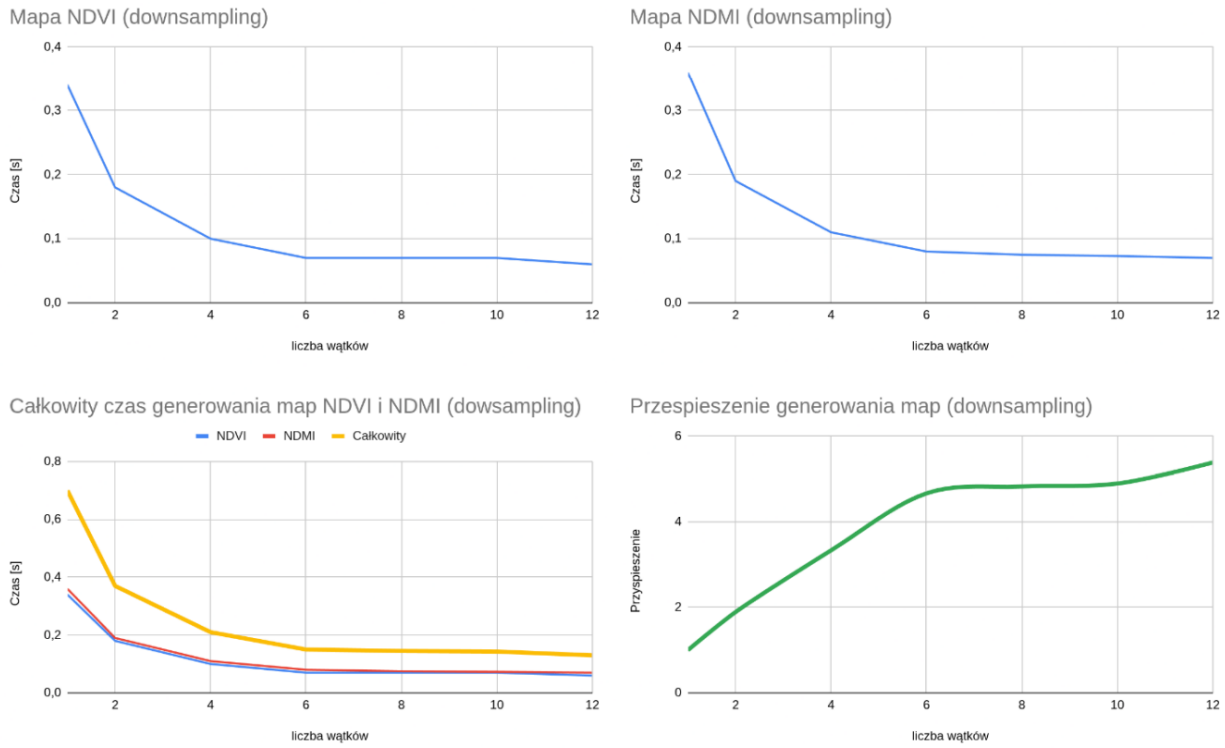
Uzyskaliśmy przyspieszenie równe 454,55% przy użyciu 12 wątków. Ponownie uzyskaliśmy mniejsze przyspieszenie w porównaniu do upsamplingu.

Generowanie mapy



Rysunek 7 Zestawienie czasu wykonania oraz przyspieszenie generowania mapy (upsampling)

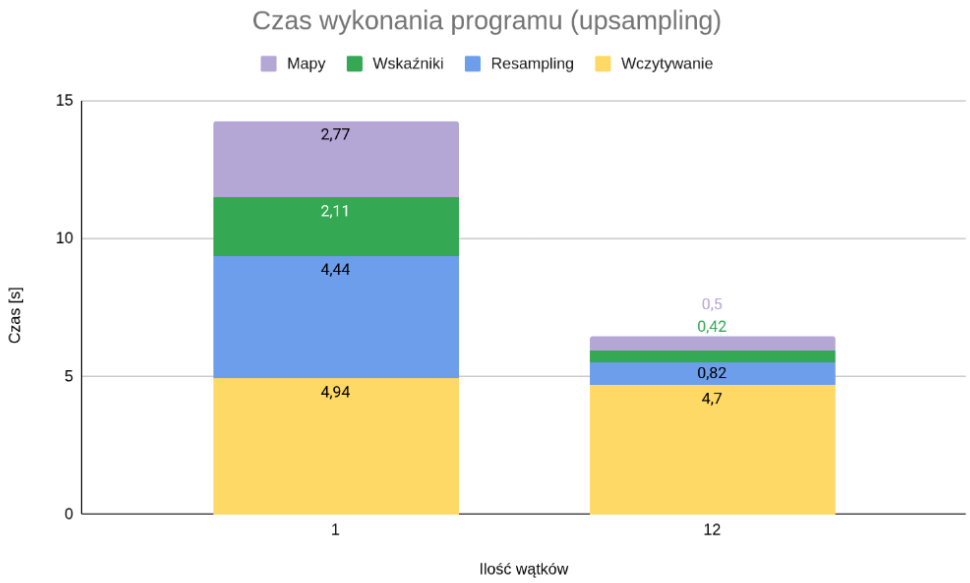
Uzyskaliśmy przyspieszenie równe 554% przy użyciu 12 wątków. Ponownie nie widać znacznej poprawy przyspieszenia powyżej 6 wątków.



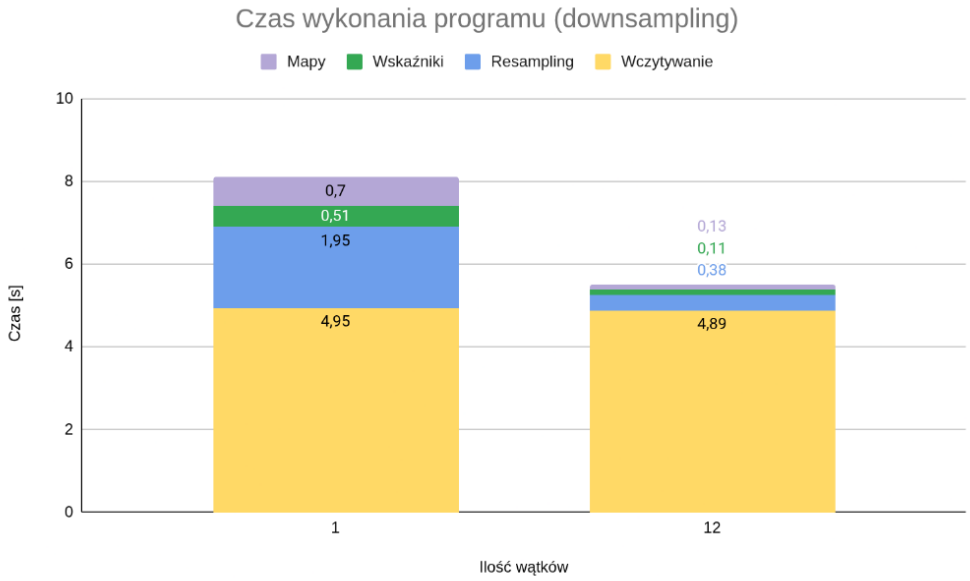
Rysunek 8 Zestawienie czasu wykonania oraz przyspieszenie generowania mapy (downsampling)

Uzyskaliśmy przyspieszenie równe 538,46% przy użyciu 12 wątków. Jest ono na podobnym poziomie, co upsampling.

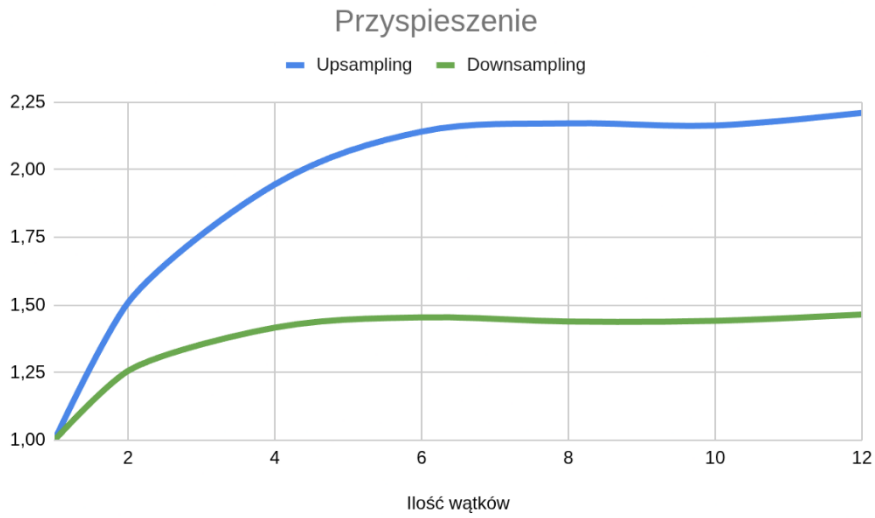
Przyspieszenie całkowite



Rysunek 9 Całkowity czas wykonania programu (upsampling)



Rysunek 10 Całkowity czas wykonania programu (downsampling)



Rysunek 11 Zestawienie całkowitego przyspieszenia programu

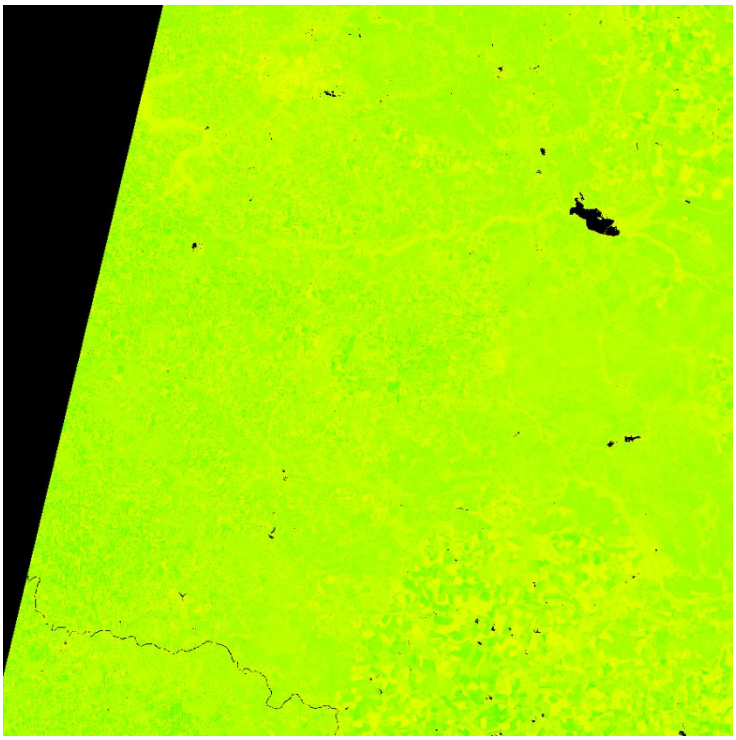
Zwiększanie liczby wątków skutecznie zmniejszało czas wykonania resamplingu, obliczania wskaźników i map, ale nie wpływało na czas wczytywania pasm. Ta operacja jest odpowiedzialna za większość czasu wykonania programu równoległego.

Upsampling posiada zauważalnie lepsze przyspieszenie, ponieważ przetwarza więcej danych (czterokrotność) w porównaniu do downsamplingu.

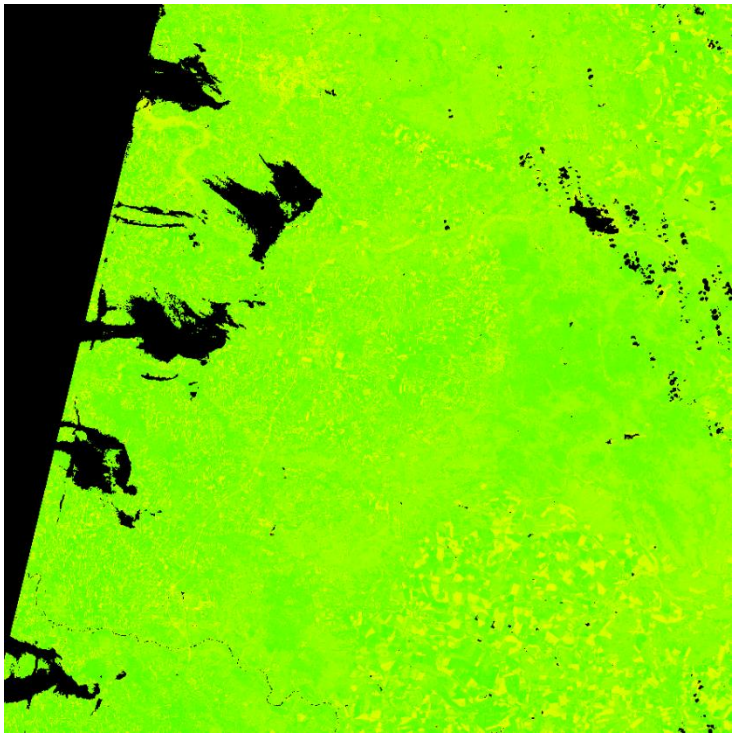
Analiza wyników programu

Program generuje mapy dwóch wskaźników teledetekcyjnych: NDVI oraz NDMI. Umożliwia on zarówno ich przeglądanie w module GUI, jak i pobieranie w formie plików `.png` na dysk komputera.

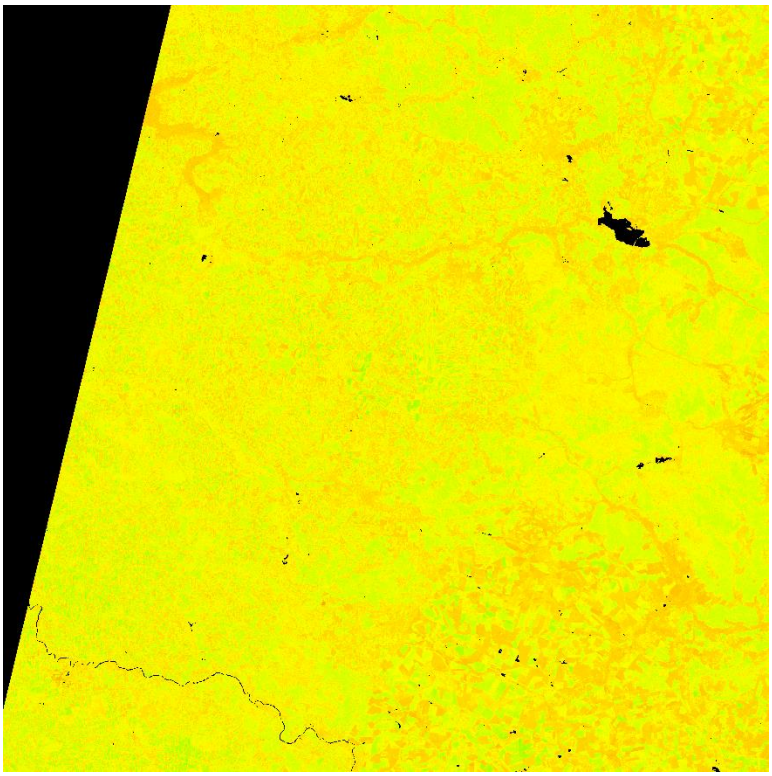
Poniżej zaprezentowano przykładowe mapy wskaźników NDVI i NDMI, opracowane dla okolic Białegostoku na podstawie zobrazowań satelitarnych Senteniel-L2a zarejestrowanych w okresach zimowym oraz wiosennym:



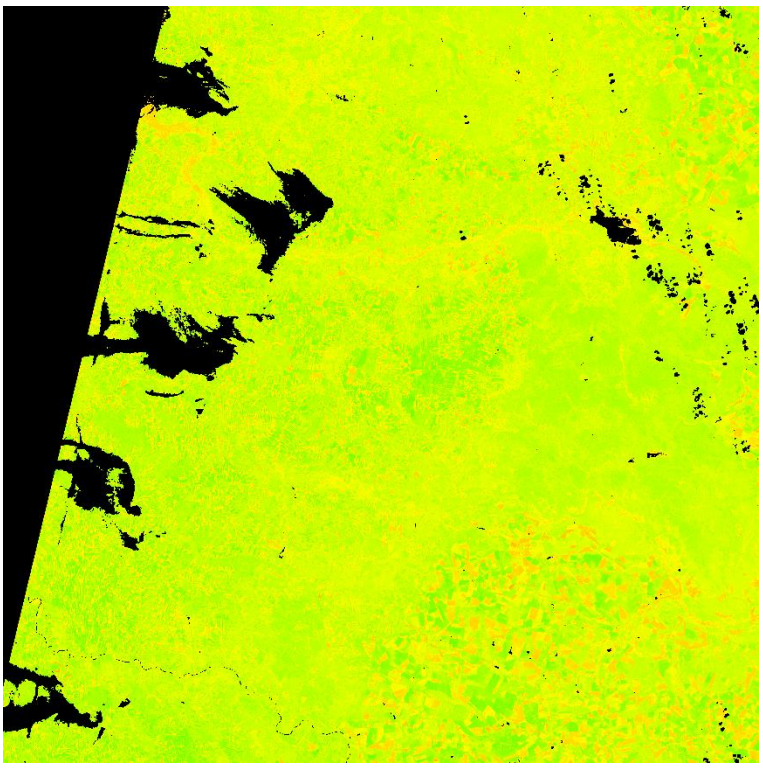
Rysunek 12 NDVI - Białystok zima



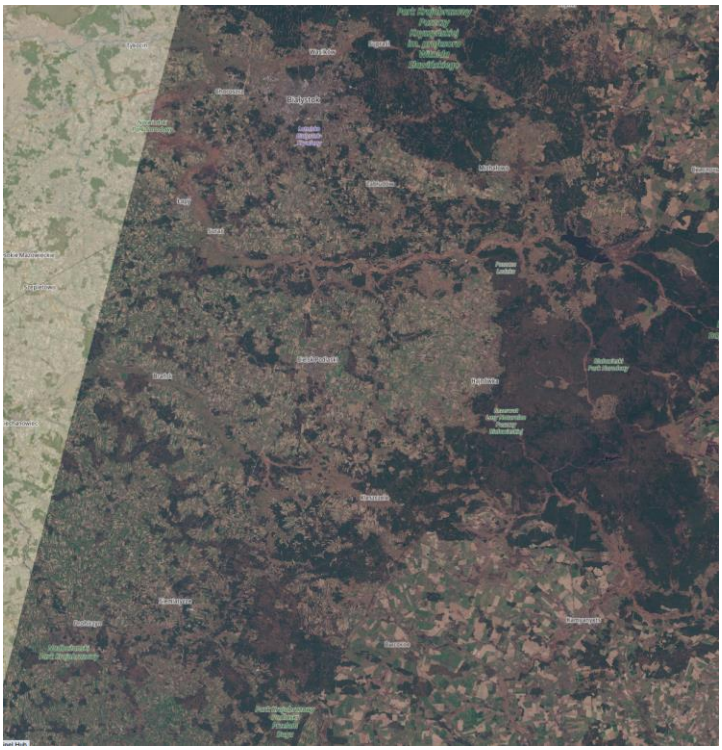
Rysunek 13 NDVI - Białystok wiosna



Rysunek 14 NDMI - Białystok zima



Rysunek 15 NDMI - Białystok wiosna



Rysunek 16 Zdjęcie satelitarne Białystok zima

Rysunek 17 Zdjęcie satelitarne Białystok wiosna

Mapy powyżej pokazują, jak zmienia się roślinność i jej nawodnienie w okolicach Białegostoku w różnych porach roku.

- **NDVI (zieleń):** Mówi, ile jest zdrowej, zielonej roślinności. Im bardziej intensywny zielony kolor, tym więcej rosnących roślin.
- **NDMI (woda):** Pokazuje, ile wody jest w liściach roślin. Im bardziej intensywny zielony, tym lepiej nawodnione są rośliny.

Z map można wywnioskować, że zimą roślinność znajduje się w stanie spoczynku. Chociaż jej ilość jest mniejsza niż wiosną, nadal utrzymuje się na dobrym poziomie. Jej spoczynek objawia się niskimi wartościami wskaźniku NDMI, co potwierdza ograniczoną zawartość wody w roślinach oraz niska wilgotność podłoża, co jest typowe dla tej chłodnej pory roku. Wiosną, jak ukazuje mapa NDMI, następuje wyraźny wzrost wilgotności w glebie i tkankach roślinnych, co jest kluczowe dla przerwania spoczynku zimowego i rozkwitu, co skutkuje zwiększeniem szaty roślinności i bardziej intensywnym zielonym kolorem na mapie NDVI.

Wnioski

Zastosowanie podejścia równoległego przyniosło duże korzyści w postaci skrócenia czasu wykonywania programu. Wybór biblioteki OpenMP do przetwarzania równoległego okazał się świetną decyzją. Ze względu na jej prostotę i łatwość implementacji, zrównoleglenie sekwencyjnego programu było trywialnym zadaniem.

W etapach takich jak: resampling, obliczanie wartości wskaźników i generowanie map wynikowych, zaobserwowano znaczne przyspieszenia, często przekraczające pięciokrotność, przy użyciu 12 wątków (np. przyspieszenie 554% dla generowania mapy dla upsamplingu oraz ponad 500% dla różnych operacji resamplingowych).

Niestety proces wczytywania pasm satelitarnych nie wykazał oczekiwanego skrócenia czasu pomimo zrównoleglenia. Jest to najprawdopodobniej efekt ograniczeń sprzętowych, w szczególności dysku twardego, który jest w stanie przetwarzać tylko jeden plik na raz. Ten czynnik miał istotny wpływ na całkowite przyspieszenie aplikacji, które dla całego procesu w trybie upsamplingu osiągnęło poziom około 2,2x, a w trybie downsamplingu około 1,4x-1,5x przy wykorzystaniu 12 wątków. Większa ilość wątków nie miałaby wpływu na zwiększenie przyspieszenia, gdyż system komputerowy, na którym uruchamiany był program ma 12 rdzeni. Ograniczenia sprzętowe można by ominąć uruchamiając program na 4 komputerach, z których każdy wczytywałby jedno pasmo, lub posiadając 4 dyski, każdy odpowiedzialny za swoje pasmo. Zastosowanie przetwarzania rozproszonego (4 lub więcej komputerów)

mogłoby nie przynieść aż takich dużych korzyści ze względu na narzuty związane z komunikacją między procesami.

Zakładając, że operacji wczytywania pasm nie można zrównoleglić, oraz jest to jedyna sekwencyjna część programu, to stanowi ona:

- 34,22% wykonania programu (upsampling)
- 60,86% wykonania programu (downsampling)

Zgodnie z prawem Amdhala:

$$S(n,p) = \frac{1}{\beta(n) + \frac{1-\beta(n)}{p}} \; dla \; K(n,p) = 0)$$

Więc

$$S(n,p) = \frac{1}{\beta(n)} \; dla \; p \to \infty$$

Zatem maksymalne przyśpieszenie dla upsamplingu wynosi:

$$S_{\max\_up} = \frac{1}{0{,}3422} = 2{,}92$$

Dla downsamplingu:

$$S_{\max\_down} = \frac{1}{0{,}6086} = 1{,}64$$

Maksymalne przyśpieszenie dla downsamplingu jest mniejsze, ponieważ część równoległa programu w tym przypadku wykonuje się szybciej niż dla upsamplingu. Te wartości przyśpieszenia pokrywają się z uzyskanym wykresem faktycznego przyspieszenia programu.