

Emil KIELAR¹

Opiekun naukowy: Sławomir HERMA², Dawid KOTRYS³

DOI: <https://doi.org/10.53052/9788366249868.10>

ANALIZA ZDJĘĆ PRZY POMOCY BIBLIOTEKI OPENCV

Streszczenie: Artykuł ten opisuje działanie programu napisanego w języku Python, który dokonuje analizy zdjęć przy pomocy biblioteki funkcji OpenCV. Kod wczytuje zdjęcie Ziemi wykonane z kosmosu i wyznacza znormalizowany różnicowy wskaźnik wegetacji. Następnie wyodrębnia z oryginalnego zdjęcia warstwy stanowiące chmury oraz ocean i przedstawia je na odrębnych obrazach.

Słowa kluczowe: program, oprogramowanie, analiza, zdjęcie, Python, OpenCV, Matplotlib, Numpy, NDVI, separacja, kolor

IMAGE ANALYSIS WITH THE OPENCV LIBRARY

Summary: This article explains a program written in Python, which analyzes images using the OpenCV function library. The code loads the photo of Earth taken from the Space and calculates NDVI. Then, the program extracts from the original photo layers that represent clouds and ocean and presents them in separate images.

Keywords: program, software, analyzes, image, Python, OpenCV, Matplotlib, Numpy, NDVI, color, separation

1. Wstęp

Analiza zdjęć jest przydatnym narzędziem w wielu dziedzinach. Jest ona używana zarówno w codziennym życiu - podczas rozpoznawania twarzy przez aparat w telefonie, jak i w specjalistycznych badaniach naukowych takich jak analiza zdjęć powierzchni Ziemi. W tym artykule zostały opisane dwa sposoby analizy obrazów: podział zdjęcia na kanały w celu obliczenia znormalizowanego różnicowego wskaźnika wegetacji (NDVI) i nałożenie na zdjęcie mapy koloru oraz separacja

¹ V Liceum Ogólnokształcące w Bielsku-Białej, emil.kielar@icloud.com

² dr inż., Katedra Inżynierii Produkcji, Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, sherma@ath.bielsko.pl

³ dr, Katedra Matematyki, Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, dkotrys@ath.bielsko.pl

kolorów ze zdjęcia. Do analizy zdjęć wykorzystywana jest biblioteka funkcji OpenCV [1]. Program został napisany w języku Python.

2. Znormalizowany różnicowy wskaźnik wegetacji

Pierwszym sposobem analizy zdjęć, przedstawionym w tym artykule, jest podział zdjęcia na kanały w celu obliczenia znormalizowanego różnicowego wskaźnika wegetacji (NDVI) oraz nałożenie na zdjęcie mapy koloru w celu lepszego zobrazowania wskaźnika.

2.1 Definicja

Znormalizowany różnicowy wskaźnik wegetacji jest graficznym wyznacznikiem umożliwiającym określenie stanu zdrowia roślinności. Najczęściej zdjęcia do analizy pochodzą z satelity krążącego wokół Ziemi. Wskaźnik został opracowany przez J. W. Rouse'a. Wyrażony jest następującym wzorem:

$$NDVI = \frac{(NIR-Red)}{(NIR+Red)} \quad (1)$$

gdzie Red jest odbiciem w paśmie czerwieni, a NIR jest odbiciem w paśmie podczerwieni. Wskaźnik NDVI przyjmuje wartości od -1 do 1. Im wyższa wartość wskaźnika, tym większa i bardziej bujna roślinność znajduje się w tym obszarze. Wskaźnik NDVI wykorzystywany jest w rolnictwie do monitorowania zdrowia plonów oraz do zbierania informacji o natężeniu fotosyntezy.

2.2 Wyznaczanie znormalizowanego różnicowego wskaźnika wegetacji

Aby wyznaczyć wskaźnik NDVI w pierwszej kolejności należy przekonwertować zdjęcie na tablicę wartości kolorów BGR. Wykorzystywana jest do tego funkcja biblioteki Numpy [5] `np.array(image, type)`.

```
import cv2
import numpy as np

image = cv2.imread("image.jpg")

image_array = np.array(image, dtype=float)
```

Do obliczenia wskaźnika NDVI zdjęcie powinno mieć duży kontrast, dlatego przed obliczeniem NDVI należy go zwiększyć.

```
def increase_contrast(image):
    max_image = np.percentile(image, 95)
    min_image = np.percentile(image, 5)

    max_bgr = 255.0
    min_bgr = 0.0

    image_ic = image - min_image
    image_ic = image_ic * ((min_bgr - max_bgr) / (min_image -
max_image))
```

```
image_ic = image_ic + min_image
```

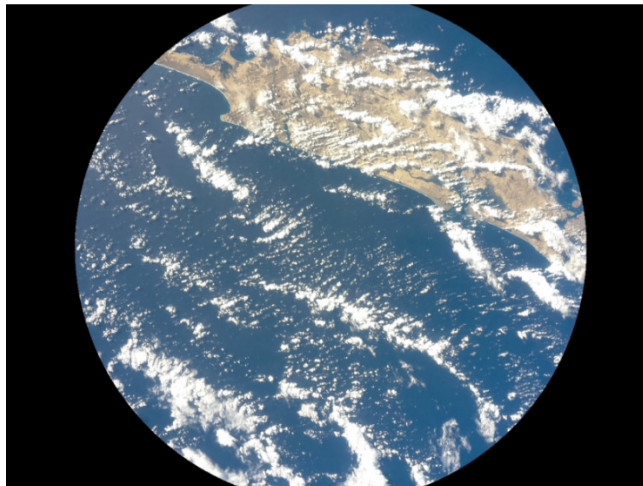
```
return image_ic
```

Zadaniem powyższej funkcji jest wyznaczenie 5 oraz 95 centylu wartości BGR z tablicy i przeliczenie wartości wszystkich elementów tak, aby zakres wartości w tablicy zawierał się w przedziale [0, 255].

Na poniższych rysunkach (rys. 1 i 2) został przedstawiony wynik działania funkcji na zdjęciu Ziemi wykonanym z pokładu Międzynarodowej Stacji Kosmicznej⁴.



Rysunek 1. Oryginalne zdjęcie



Rysunek 2. Zdjęcie ze zwiększonym kontrastem

Po zwiększeniu kontrastu zdjęcia, można przystąpić do obliczenia wskaźnika NDVI. Zastosowany zostanie wzór z podrozdziału 2.1.

⁴ Zdjęcie zostało wykonane podczas działania programu autora artykułu na Międzynarodowej Stacji Kosmicznej w ramach konkursu „The European Astro Pi Challenge”. <https://astro-pi.org>

```
b, g, r = cv2.split(image_ic)
denominator = (r.astype(float) + b.astype(float))
denominator[denominator==0] = 0.01

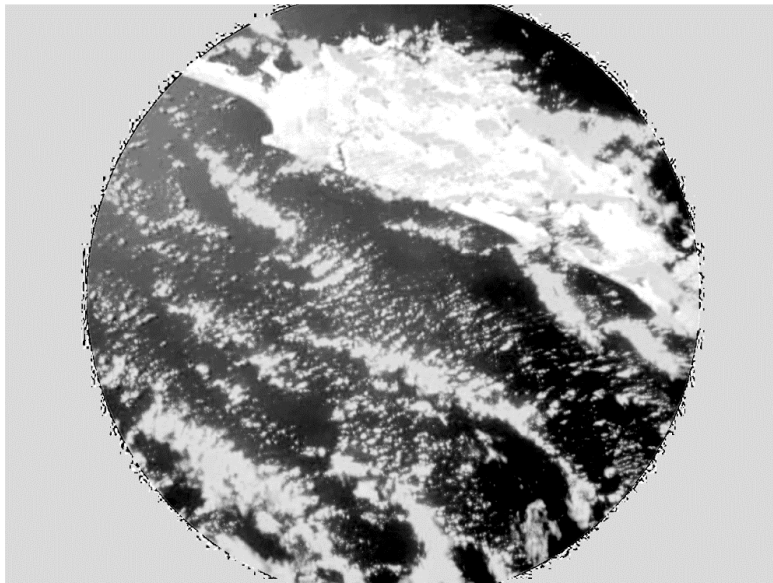
numerator = (r.astype(float) - b.astype(float))
ndvi = numerator/denominator

ndvi_ic = increase_contrast(ndvi)

cv2.imwrite("image-ndvi-greyscale.png", ndvi_ic)
```

W programie następuje podział tablicy *image_ic* na trzy kanały. Następnie mianownik jest obliczany według wzoru. W celu uniknięcia błędu dzielenia przez zero, wartości mianownika równe zero są zamieniane na wartość 0,01. Następnie obliczane są wartości licznika wzoru oraz wyznaczany jest wskaźnik NDVI.

Na obliczonym wskaźniku NDVI wykonywana jest jeszcze raz operacja zwiększenia kontrastu. W jej wyniku otrzymywany jest wskaźnik NDVI w skali szarości. Jaśniejsza barwa piksela oznacza wyższą wartość wskaźnika NDVI.



Rysunek 3. Zdjęcie z wyznaczonym wskaźnikiem NDVI w skali szarości

2.3 Nałożenie na zdjęcie mapy koloru

Obliczony wskaźnik NDVI na rysunku 3 jest mało czytelny. W celu jego poprawy można zastosować mapę koloru.

Intensywność wskaźnika została przekształcona na barwy za pomocą bibliotek funkcji OpenCV oraz Matplotlib⁵ z użyciem mapy koloru *viridis* [3].

```
from PIL import Image
import matplotlib.pyplot as plt
import cv2
import numpy as np

def get_colormap(cmap_name):
    cmap = plt.get_cmap(cmap_name)

    sm = plt.cm.ScalarMappable(cmap=cmap)

    color_range = sm.to_rgba(np.linspace(0, 1, 256),
bytes=True)[:,:-1]

    return color_range.reshape(256, 1, 3)

image_ndvi = cv2.imread("image-ndvi-greyscale.png")

# Nałożenie na zdjęcie mapy koloru
img_ndvi = cv2.applyColorMap(image_ndvi,
get_colormap('viridis')).astype(np.int)

cv2.imwrite("image-ndvi.jpg", img_ndvi)

# Usunięcie ze zdjęcia NDVI wskaźnika na czarnym tle
image_ndvi = cv2.imread("image-ndvi.jpg")
image_ndvi_array = np.array(image_ndvi, dtype=float)

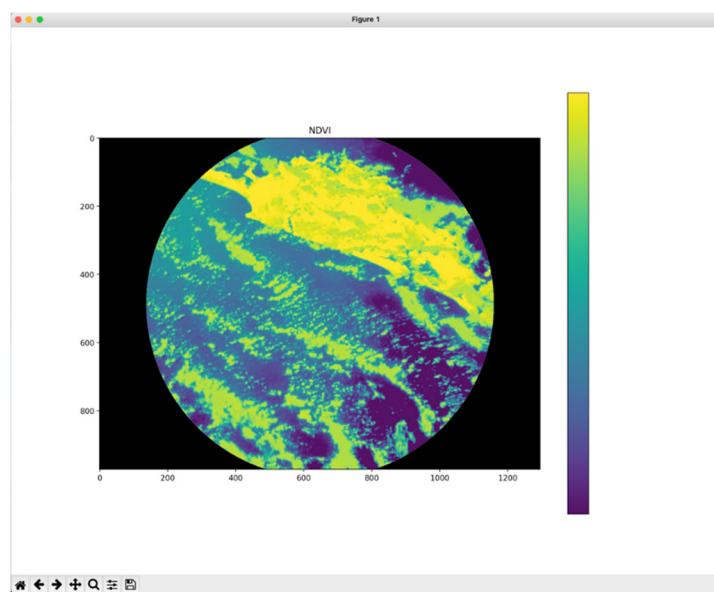
mask_window = np.zeros((height, width, 1), np.uint8)
mask_window = cv2.circle(mask_window, (int(width/2),
int(height/2)), int(height/1.9), (255), -1)

image_ndvi = cv2.bitwise_and(image_ndvi_array,
image_ndvi_array, mask = mask_window)
cv2.imwrite("image-ndvi.jpg", image_ndvi)

#Wyświetlenie obrazu z legendą
img = Image.open("image-ndvi.jpg")
im_show = plt.imshow(img)
plt.colorbar(ticks=[])
plt.title("NDVI")
plt.show()
```

Program importuje biblioteki Matplotlib, OpenCV oraz Numpy. Następnie funkcja *get_colormap* pobiera mapę koloru z biblioteki Matplotlib i nakłada ją na zdjęcie *image-ndvi-greyscale.png*. Zdjęcie z nałożoną mapą koloru jest zapisywane jako *image-ndvi.jpg*. Wykorzystując funkcję *plt.show()* w nowym oknie prezentowany jest obraz z legendą.

⁵ <https://matplotlib.org>



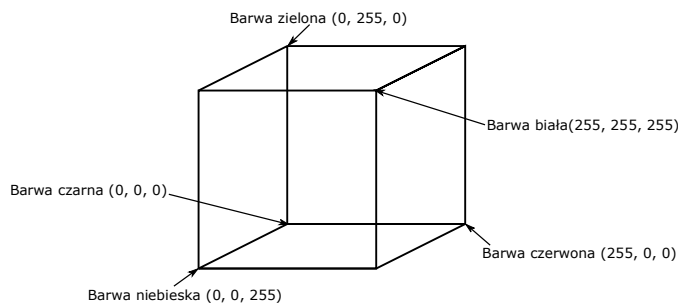
Rysunek 4. Okno ze zdjęciem NDVI oraz legendą wskaźnika NDVI

3. Separacja kolorów ze zdjęcia

Drugim sposobem analizy zdjęć, przedstawionym w tym artykule, jest separacja kolorów białego i niebieskiego ze zdjęcia. Celem tej operacji jest odseparowanie warstwy chmur i powierzchni oceanu ze zdjęcia Ziemi wykonanego z kosmosu.

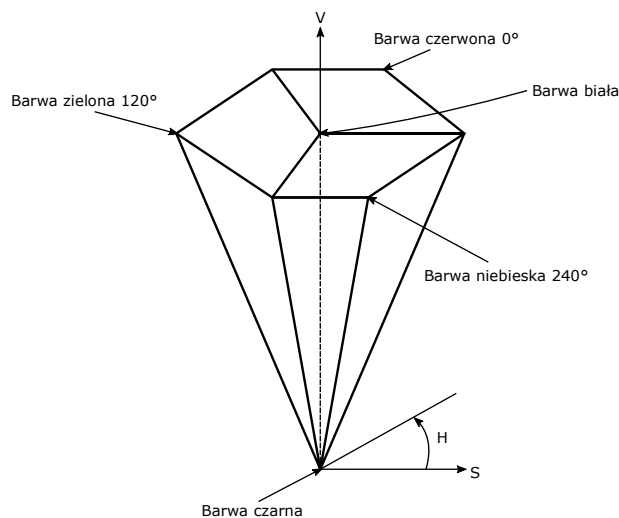
3.1 Sposoby zapisu kolorów w tablicy

Model barw RGB jest najczęściej stosowanym i spotykanym modelem barw. Wykorzystuje się go w monitorach oraz programach do edycji grafiki rastrowej. Barwami podstawowymi są **R** (barwa czerwona), **G** (barwa zielona) oraz **B** (barwa niebieska). Barwy są mieszane *addytywnie*. Oznacza to, że dowolna barwa w modelu RGB jest sumą udziałów poszczególnych barw podstawowych. Sześcian RGB przedstawia sposób otrzymywania kolorów.



Rysunek 5. Sześcian RGB

Model barw HSV jest modelem zorientowanym na użytkownika, w przeciwieństwie do modelu RGB ukierunkowanego sprzętowo. Do wyrażenia koloru wykorzystywane są trzy zmienne: **H** (odcień barwy), **S** (nasycenie koloru), **V** (moc światła białego). Do przedstawienia tej przestrzeni barw jest stosowany stożek, którego podstawą jest sześciokąt foremny. H jest wyrażone jako kąt obrotu wokół osi prostopadłej i przechodzącej przez środek symetrii sześciokąta foremnego. Barwa czerwona ma kąt 0° , barwa zielona 120° a barwa niebieska 240° . Wartość S to ułamek zmieniający wartości od 0 do 1 na osi i bokach ostrosłupa. V to zmienna przyjmująca wartości z przedziału $[0, 1]$ wzdłuż wysokości danej figury. Najmniejszą wartość przyjmuje na podstawie ostrosłupa, a największą w wierzchołku.



Rysunek 6. Ostrosłup HSV

Przedział koloru można precyzyjniej określić przy pomocy modelu barw HSV, dlatego został on zastosowany w funkcji separującej kolor ze zdjęcia.

3.2 Funkcja separująca kolor ze zdjęcia

Wyodrębnienie kolorów ze zdjęcia odbywa się przy pomocy funkcji z biblioteki OpenCV. Jej zadaniem jest identyfikacja kolorów należących do wskazanego przedziału i przeniesienie ich do osobnego obrazu.

```
import cv2
import numpy as np

image = cv2.imread("image.jpg")

image_hsv = cv2.cvtColor(image_window, cv2.COLOR_BGR2HSV)

#Blue
```

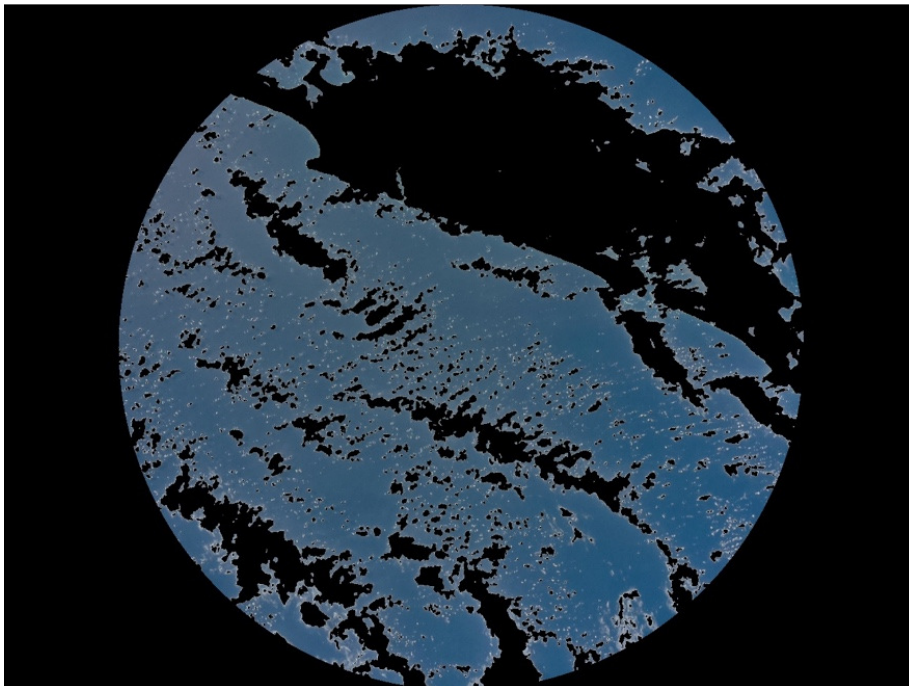
```
lower_blue = np.array([75, 32, 32])
upper_blue = np.array([120, 255, 255])

mask_blue = cv2.inRange(image_hsv, lower_blue, upper_blue)
image_blue = cv2.bitwise_and(image_hsv, image_hsv, mask =
mask_blue)

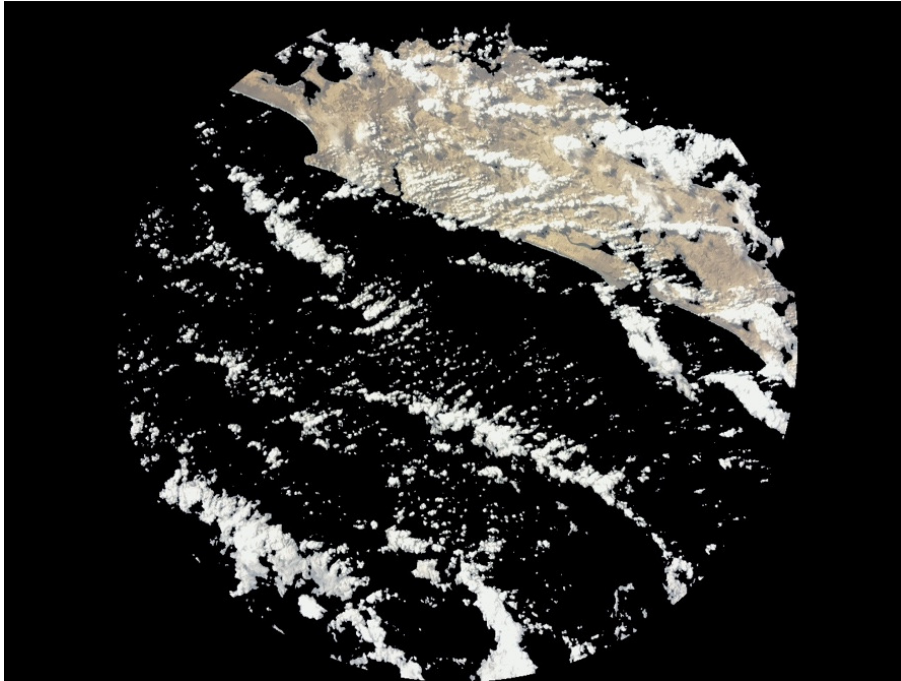
mask_nonblue = cv2.bitwise_not(mask_blue)
image_nonblue = cv2.bitwise_and(image_hsv, image_hsv, mask =
mask_nonblue)
```

W programie wczytywane jest zdjęcie i zmieniany jest jego model barw z BGR na HSV. Następnie definiowany jest przedział koloru niebieskiego: jego dolna (*lower_blue*) oraz górna (*upper_blue*) granica. Kod wyszukuje wszystkie piksele należące do przedziału i zapisuje je w masce *mask_blue*. Następnie tworzone jest zdjęcie z zaznaczonymi wyłącznie pikselami niebieskimi (*image_blue*). Na koniec program tworzy obraz z pikselami, które nie są koloru niebieskiego (*image_nonblue*).

Przykładowe zdjęcia z zaznaczonymi pikselami niebieskimi i z pikselami, które nie są koloru niebieskiego:



Rysunek 7. Zdjęcie z zaznaczonymi pikselami niebieskimi



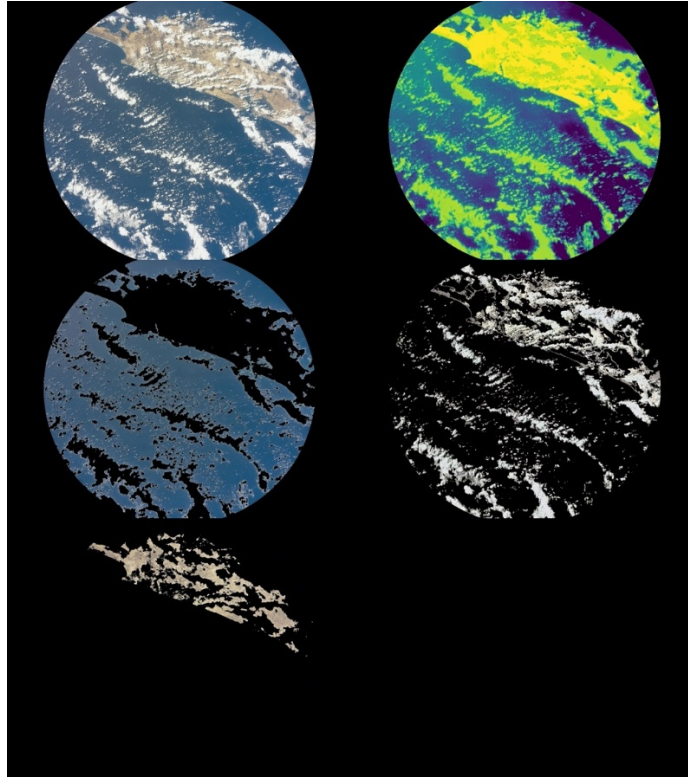
Rysunek 8. Zdjęcie z zaznaczonymi pikselami, które nie są koloru niebieskiego

Odseparowanie innych kolorów można zrealizować przez zmianę górnej i dolnej granicy przedziału koloru.

4. Podsumowanie

Opisane algorytmy zostały wykorzystane przez autora artykułu w projekcie Astro Pi Challenge, którego celem była identyfikacja rodzaju obszaru, nad którym znajduje się Międzynarodowa Stacja Kosmiczna poprzez separację warstw ze zdjęcia Ziemi na ocean, chmury i ląd oraz określenie możliwości rozwoju roślinności za pomocą wskaźnika NDVI. Na poniższym rysunku został przedstawiony wynik działania dla przykładowego zdjęcia.

Zaprezentowane funkcje można zastosować w wielu obszarach naukowych, takich jak meteorologia, ochrona środowiska, rolnictwo czy medycyna.



Rysunek 9. Kompozycja złożona ze zdjęcia oryginalnego oraz graficznego przedstawienia wskaźnika NDVI i odseparowanych warstw chmur, oceanu i lądu.

LITERATURA

1. Dokumentacja biblioteki funkcji OpenCV [online]. <https://opencv.org>, 18.10.2021
2. Normalized difference vegetation index [online]. Wikipedia: the free encyclopedia. 18.10.2021
3. Serwis internetowy: https://en.wikipedia.org/wiki/Normalized_difference_vegetation_index
4. Choosing Colormaps in Matplotlib [online]. Dokumentacja biblioteki funkcji Matplotlib. 18.10.2021 <https://matplotlib.org/stable/tutorials/colors/colormaps.html>
5. Capture plant health with NDVI and the Raspberry Pi [online]. Raspberry Pi foundation. 19.10.2021
6. Serwis internetowy: <https://projects.raspberrypi.org/en/projects/astropi-ndvi/3>
7. FOLEY J. D., VAN DAM A., FEINER S. K., HUGHES J. F., PHILLIPS R. L.: Wprowadzenie do grafiki komputerowej. Wydawnictwo Naukowo-Techniczne, Warszawa 1995.
8. Serwis internetowy: <https://numpy.org>