

# Automatyzacja oceny nasilenia objawów chorobowych septorioz zbóż z wykorzystaniem komputerowej analizy obrazu w języku programowania Python

Automation of septoria disease severity assessment using digital image analysis in Python programming language

Sławomir Bartosiak ✉

Instytut Hodowli i Aklimatyzacji Roślin – Państwowy Instytut Badawczy,  
Zakład Fitopatologii, Pracownia Hodowli Odpornościowej,  
✉ e-mail: s.bartosiak@ihar.edu.pl

Ocena objawów chorobowych septorioz na liściach zbóż, opisywanie i tworzenie dokumentacji fotograficznej poszczególnych liści jest czasochłonnym i pracochłonnym zadaniem. W opracowaniu przedstawiona została automatyzacja oceny objawów chorobowych septorioz zbóż za pomocą aplikacji open-source stworzonych w języku Python. Oprogramowanie umożliwia automatyzację odczytywania nazw obiektów doświadczalnych oraz ocenę objawów chorobowych poszczególnych liści, dzięki czemu istnieje możliwość np. usuwania obserwacji odstających z analizy.

**Słowa kluczowe:** Python, analiza obrazów cyfrowych, septoriozy, liście, pszenica, pszenżyto

The assessment of the severity of symptoms of septoria disease is both time- and labour consuming. The paper presents a means of automating the evaluation of cereal septoria disease symptoms using open-source applications developed in the Python programming language. The software enables computerized documentation and reading of study subject items and the assessment of individual leaf disease symptoms, making it possible to remove outliers from the analysis.

**Key words:** Python, digital image processing, septoria, Stagonospora Nodorum Blotch, leaves, wheat, triticale

## Wstęp

Szybka ocena objawów chorobowych jest jednym z kluczowych elementów hodowli odpornościowej (Singh i Misra, 2017). Klasycznym podejściem było stosowanie oceny wizualnej w przyjętej skali. Obecnie wraz z rozwojem technik informacyjnych, fotografii cyfrowej oraz zaawansowanego obrazowania wielospektralnego odchodzi się od metod klasycznych, które są obciążone błędem spowodowanym subiektywną oceną wizualną na rzecz szybkich i dokładnych metod komputerowego przetwarzania obrazu (Thomas *i in.*, 2018). Do przetwarzania i analizy obrazu można wykorzystywać oprogramowanie komercyjne lub open source. Minusami wykorzystywania oprogramowania komercyjnego są dodatkowe koszty, które można ograniczyć wykorzystując oprogramowanie open source. Programy open source, np. ImageJ oraz ImageJ2 są pomocne, jednak wymagają one znajomości przez użytkownika graficznego interfejsu oraz dostępnych funkcji. Wprawdzie programy te

pozwalają na tworzenie makr umożliwiających automatyzację, jednak ich wykonanie jest często trudne i czasochłonne (Easlon i Bloom, 2014; Rueden *i in.*, 2017). Starsze generacje programu ImageJ nie wykorzystywały też w pełni mocy obliczeniowej nowej generacji komputerów. Rozwój języków programowania wysokiego poziomu oraz dostęp do szerokiego pakietu bibliotek standardowych w znaczący sposób ułatwił tworzenie aplikacji, w tym analizę i przetwarzanie obrazu. Jednym z języków tego rodzaju przyjaznych dla użytkownika jest Python (Easlon i Bloom, 2014).

Testy patogeniczności w warunkach kontrolowanego środowiska w stadium siewki, umożliwiają szybką ocenę objawów chorobowych, a przez to skrócenie cyklu hodowlanego. Są one szeroko stosowane w praktyce hodowlanej oraz w badaniach naukowych (Thomas *i in.*, 2018, Easlon i Bloom, 2014; Singh i Misra, 2017). Jednym z pracochłonnych zadań podczas prowadzenia doświadczeń w komorach klimatycznych

jest opisywanie i tworzenie dokumentacji fotograficznej. W niniejszej pracy omówiona zostanie metodyka umożliwiająca automatyzację odczytywania oraz generowania etykiet obiektów doświadczalnych poprzez zastosowanie kodów QR – Quick Response w omawianych aplikacjach.

W niniejszym opracowaniu termin stopień porażenia oznacza iloraz tkanki porażonej do powierzchni całkowitej (suma tkanki zdrowej i porażonej) wyrażony jako procent. W ocenie stopnia porażenia liści zbóż można stosować podejście zakładające iloraz sumy powierzchni objawów chorobowych do sumy całkowitej powierzchni liści (średnia ważona) lub średnią arytmetyczną ze stopnia porażenia poszczególnych liści. W celu obliczenia średniej arytmetycznej niezbędna jest segmentacja obrazu źródłowego na poszczególne liście. Zastosowanie średniej ważonej zwiększa wagę liści o większej powierzchni na ocenę stopnia porażenia obiektu testowego, natomiast poprzez zastosowanie średniej arytmetycznej ze stopnia porażenia poszczególnych liści ogranicza się wpływ powierzchni liści na stopień porażenia. Podejście zakładające średnią arytmetyczną może być pomocne w obiektach charakteryzujących się dużym zróżnicowaniem powierzchni liści. Celem pracy było stworzenie aplikacji w języku programowania Python przeznaczonej do oceny stopnia porażenia liści pszenicy i pszenżyta wywołanych przez patogeny nekrotroficzne *Zymoseptoria tritici* oraz *Parastagonospora nodorum*.

## Materiały i Metody

**Aplikacje dostępne są do pobrania przez platformę Github.com (<https://github.com/SlawomirBartosiak/septoria-leafQR.git>).**

W celu pobrania aplikacji należy kliknąć w link i wybrać zielony przycisk *clone or download*. Pobrany plik zip rozpakować, najlepiej bezpośrednio na dysku C: lub dowolnym katalogu z zaznaczeniem, że w ścieżce dostępu nie mogą znajdować się polskie znaki. W przeciwnym wypadku niektóre polecenia gdzie używane są ścieżki dostępu z polskimi znakami nie będą się wykonywać i program nie będzie działał poprawnie.

### **Oprogramowanie i parametry komputera, na którym wykonano testy oprogramowania:**

Windows 10, 64 – bitowy system operacyjny; procesor x64, Intel®Core™ i5–4590; pamięć ram –800Mhz, DDR, 3 8 GB; karta graficzna Intel HD Graphics 4600; dysk twardy Seagate Barracuda 500 GB 3.5” SATA III (ST500DM002), 7200 obr./min,

SATA III.

### **Doświadczenie:**

W opracowaniu wykorzystano dokumentację fotograficzną z doświadczeń dotyczących septorioz zbóż wykonanych w komorach klimatycznych. Obiekty testowe były wysiane w wielodoniczkach – 8 x 13 komórek na palecie. Temperatura w komorze klimatycznej wynosiła 22/20°C przez 16/8 godzin. Oświetlenie 16/8 godzin. Po 14 dniach od wysiania obiekty inokulowano wodną zawiesiną zarodników *P. nodorum* o stężeniu 4 mln x ml<sup>-1</sup>. Porażone liście 3-tygodniowych siewek pszenicy i pszenżyta zostały przyklejone na przezroczystą folię przyklepną, a następnie etykietowane i fotografowane w namiocie bezcieniowym. Do oceny wykorzystano drugi liść. Odmiany pszenicy i pszenżyta w doświadczeniu charakteryzowały się zróżnicowaną podatnością na *P. nodorum*.

### **Przygotowanie dokumentacji fotograficznej (Rys. 1):**

1. Etykieta – zawierająca kod QR oraz nazwę obiektu testowego.
2. Niebieskie tło – cyjan.
3. Porażone liście na płaskiej powierzchni równomiernie przylegające do powierzchni.
4. Równomierne oświetlenie fotografowanego obiektu w namiocie bezcieniowym – temperatura barwowa 5500K, 2x 6055Lm.
5. Aparat lub inne urządzenie optoelektryczne ustawione prostopadle do fotografowanej powierzchni, najlepiej na sztywnym statywie, rejestrujące obrazy w przestrzeni kolorów sRGB w rozdzielczości przynajmniej 3072x2048 (180dpi), głębia koloru – 24 bitowa, jednostka przesłony – f/4, czas ekspozycji – 250 s, szybkość ISO – ISO100, długość ogniskowej 50mm, bez lampy błyskowej.

### **Wymagania:**

- Windows 10;
- Python 3.8.2 (<https://www.python.org/downloads/windows/>) – w przypadku gdy użytkownik chce uruchamiać programy o rozszerzeniu \*.py w wierszu polecenia Windows, przy instalacji należy dodać Pythona do ścieżki zaznaczając opcję „Add Python 3.8 to PATH”;
- zainstalowane moduły Python:
  - pip 19.2.3
  - opencv-python 4.2.0.32
  - numpy 1.18.1
  - pandas 1.0.3
  - pyzbar 0.1.8
  - pypng 0.0.20

- Pillow 7.1.2
- pyqrcode 1.2.1

#### **Aplikacja *lab\_gener.py*:**

W celu ułatwienia rozpoznawania etykiet z plików graficznych przez oprogramowanie zastosowano kody QR. Aplikacja *lab\_gener.py* służy do generowania kodów QR z listy obiektów wpisanych w pliku tekstowym *object\_list.txt*.

#### **Przygotowanie do pracy:**

1. Przygotować listę obiektów i zapisać do pliku *object\_list.txt* w katalogu głównym *septoria-leafQR*. Każdy kod obiektu powinien znajdować się w nowym wierszu:  
Object 1, Repetition 1  
Object 1, Repetition 2  
Object 2, Repetition 1  
Itd.
2. Następnie należy uruchomić aplikację, w wierszu poleceń Windows należy przejść do katalogu *septoria-leafQR* i wpisać *python lab\_gener.py* lub dwukrotnie kliknąć na program jeśli domyślnie otwierany jest w aplikacji Python3.
3. Wyświetli się komunikat „Please enter [y] – yes, to add titles to output files or [n] – no, to save files without title” w celu wyboru opcji z zapisem lub bez zapisu tytułów wprowadzonych etykiet w wygenerowanych plikach graficznych.

4. Po wpisaniu „y” lub „n” program rozpocznie pracę wyświetlając nazwy zapisanych plików graficznych w katalogu *labels\_output*.
5. Po zakończonej pracy wyświetli się komunikat *Finished*.

Etykiety należy wydrukować, najlepiej na niebieskim (cyjan) tle i umieścić na folii przylepnej z liśćmi i wykonać fotografię lub skan (Rys.1).

#### **Aplikacja *sleaves.py*:**

Aplikacja analizuje stopień porażenia obiektów doświadczalnych przygotowanych według powyższej procedury. Program pobiera pliki graficzne do analizy z katalogu *input\_images*. Etykiety obiektów doświadczalnych są automatycznie rozpoznawane przez program z kodów QR. W przypadku gdyby kod QR nie został odczytany przez program, np. z powodu rozmazanego zdjęcia, program wykorzysta nazwę odczytanego pliku graficznego. Program pobiera pliki do analizy z katalogu *input\_images*. Program posiada możliwość zapisu plików graficznych z objawami porażenia oraz poszczególnych liści. Użytkownik może również modyfikować parametry odcieni – HUE w przestrzeni barw HSV, w celu optymalizacji określania powierzchni porażonej oraz całkowitej (zdrowej tkanki + porażonej tkanki) w przypadku gdy parametry używanego sprzętu różnią się od tych opisanych w omawianej metodyce.



Rys. 1. Przykład przygotowanej dokumentacji fotograficznej obiektu doświadczalnego.  
Fig. 1. Example of prepared image sample documentation to analyse.

**Optymalizacja parametrów:**

1. Pliki graficzne przygotowane według powyższej metodyki należy umieścić w katalogu *input\_images*.
2. W celu aktywacji zapisu wyciętych konturów liści należy zmienić wartość zmiennej *save\_leaf\_im* na *True* w pliku *sleaves.py*.  
`save_leaf_im = True # True/False save an extracted leaf image`
3. W celu zapisu porażonego obszaru poszczególnych liści należy zmienić wartość zmiennej *save\_diseased\_im* na *True* w pliku *sleaves.py*.  
`save_diseased_im = True # True/False save an diseased image`
4. Zapisane obrazy będą się znajdować w katalogu *output\_analysis*.
5. Następnie należy uruchomić program. W wierszu poleceń Windows należy przejść do katalogu *septoria-leaf* i uruchomić program wpisując *python sleeves.py* lub przez podwójne kliknięcie ikony.
6. Jeśli program przycina zbyt mocno kontury liści należy zmienić zakres HUE\* w zmiennych *leaf\_hue\_min* i *leaf\_hue\_max* na szerszy, a jeśli w konturach liści znajduje się tło należy zmniejszyć zakres. Domyślnie wynosi on 0 do 90.  
`leaf_hue_min = 0 # Min leaf HUE, default 0`  
`leaf_hue_max = 90 # Max leaf HUE, default 90`
7. Jeśli program przycina zbyt mocno objawy septorioz należy zwiększyć zakres HUE\*, a jeśli uchwycona jest też zdrowa tkanka należy zmniejszyć zakres.  
`diseased_hue_min = 0 # Min diseased HUE, default 0`  
`diseased_hue_max = 45 # Max diseased HUE, default 45`
8. Po optymalizacji ustawień należy ponownie uruchomić program i sprawdzić czy optymalizacja parametrów przebiegła pomyślnie.

\* zakres odcieni – HUE jest wyrażony w skali HSV, a przeliczany jest ze skali sRGB. Odcień w przestrzeni kolorów HSV w bibliotece *Open CV* może mieścić się w zakresie od 0 do 179.

**Przygotowanie do pracy:**

1. Jeśli przygotowana dokumentacja fotograficzna doświadczenia znajduje się w katalogu *input\_images* w wierszu poleceń Windows należy przejść do katalogu *septoria-leaf* i uruchomić program wpisując *python sleeves.py* lub przez podwójne kliknięcie.
2. Program rozpocznie analizę trwającą około 20 sekund na 100 zdjęć (czas może się różnić

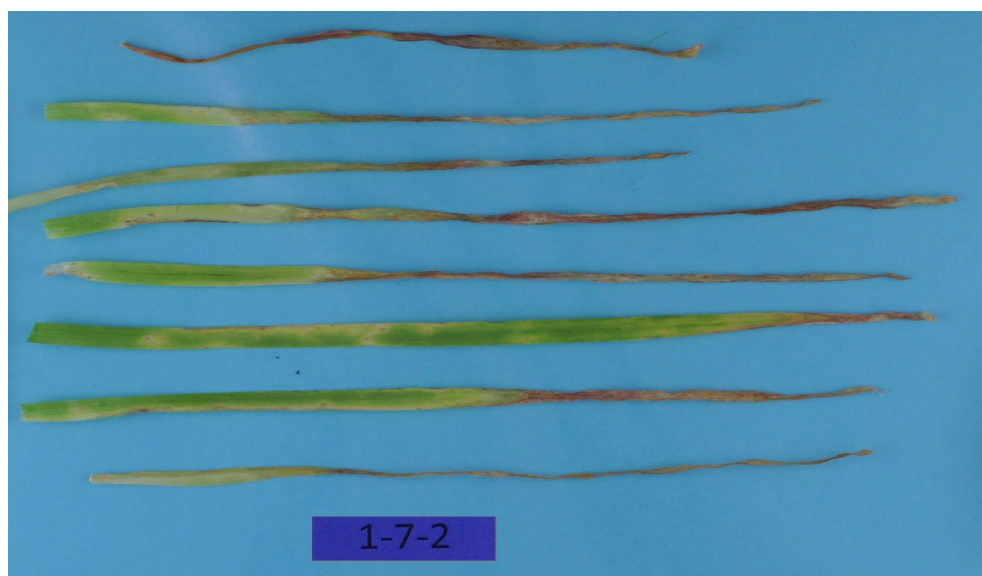
w zależności od mocy obliczeniowej komputera oraz liczby zdjęć do analizy).

3. Po wykonaniu operacji pojawi się komunikat *Finished* i w katalogu głównym *septoria-leaf* zapisane zostaną podsumowania w formacie \*.csv:
  - a. *results.csv* – zawierający wyniki analizy poszczególnych liści:
    - *sample\_name* – nazwa obiektu,
    - *leaf\_area* – powierzchnia wyciętego liścia w pikselach,
    - *diseased\_area* – powierzchnia porażonej tkanki liścia w pikselach,
    - *diseased\_percent* –  $(\text{diseased\_area}/\text{leaf\_area}) \cdot 100$ .
  - b. *pivot\_table.csv* – zawierający tabelę przedstawiającą wyniki analizy poszczególnych obiektów:
    - *sample\_name* – nazwa obiektu,
    - *leaf\_area* – suma powierzchni liści danego obiektu w pikselach,
    - *diseased\_area* – suma powierzchni porażonej danego obiektu w pikselach,
    - *diseased\_percent* – średnia arytmetyczna ze stopnia porażenia poszczególnych liści danego obiektu.

**Wyniki i Dyskusja**

Aplikacja *sleaves.py* wykorzystuje w pełni dostępne zasoby CPU. Analiza 130 zdjęć o rozdzielczości 3072x2048 na średniej klasy komputerze (szczegółowe parametry w rozdziale Metodyka) trwała 21 sekund. W trybie zapisu wyciętych liści i objawów ze zdjęć analiza trwała 40 sekund.

W testach z wykorzystaniem dokumentacji fotograficznej z doświadczeń w komorach klimatycznych nie wykazano różnic istotnie statystycznych porównując metodę wykorzystującą średnią ważoną i średnią arytmetyczną z poszczególnych liści (ANOVA,  $p=0.62$ ). Obserwowano różnice w statystykach opisowych. Algorytm wykorzystujący średnią arytmetyczną ze stopnia porażenia poszczególnych liści zwiększał odchylenie standardowe o 0,7% w porównaniu do metody zakładającej średnią ważoną z ze stopnia porażenia liści – (iloraz sumy powierzchni porażonej i powierzchni całkowitej). Średnie porażenie zwiększyło się o 1,27% w przypadku zastosowania średniej arytmetycznej. Największe różnice obserwowano w obiektach testowych, w których występowały duże dysproporcje pomiędzy stopniem porażenia liści i powierzchnią liści. Porażona tkanka pszenicy i pszenżyta przez septoriozy (nekrozy) w stadium siewki może powodować wyschnięcie, skręcanie i zwężanie się liścia w miejscu porażenia (Rys.2).



Rys. 2. Objawy nekroz na drugim liściu siewek pszenżyta wywołane przez *P. nodorum*.

Fig. 2. Septoria disease symptoms caused by *P. nodorum* on second leaves of triticale seedlings.

Skutkuje to zmniejszeniem powierzchni porażonej w stosunku do powierzchni zdrowej. Obiekty testowe, w których występowały duże nieporażone liście zmniejszały średnią nawet o 12%. Dysponując stopniem porażenia każdego liścia z osobna można też usunąć odstające obserwacje zwiększając dokładność doświadczenia (Rys 2. – trzeci liść od dołu widocznie obniża średnią).

Aby aplikacja działała bez zarzutu niezbędna jest optymalizacja parametrów co wymaga podstawowej znajomości języka Python oraz obsługi komputera, nie powinno być to jednak przeszkodą dla początkujących użytkowników. Omawiane aplikacje można też modyfikować i dostosowywać do własnych potrzeb. Po modyfikacji parametrów aplikację z powodzeniem można stosować do oceny innych chorób zbóż lub też innych gatunków roślin, a w przypadku modyfikacji kodu można znacznie rozszerzyć działanie aplikacji. Należy jednak zaznaczyć, że w przypadku liści pochodzących np. z doświadczeń polowych zastosowane algorytmy nie będą w stanie odróżnić objawów chorobowych powodowanych przez septoriozy od niedoborów składników pokarmowych, które również mogą powodować chlorozy lub innych chorób o podobnych objawach. Nekrozy powodowane przez septoriozy mogą nie zostać odróżnione od objawów suszy. Zaleca się więc używanie aplikacji w doświadczeniach, w których stosuje się inokulację lub w komorach klimatycznych, gdzie nie ma możliwości wystąpienia niepożądanych stresów biotycznych i abiotycznych. Aplikacja jest darmowa na licencji MIT, a więc można ją swobodnie modyfikować i wykorzystywać. Przewagą

rozwiązań komercyjnych natomiast jest bardziej intuicyjny interfejs graficzny, który jest przejrzysty i łatwiejszy dla użytkowników, wiąże się to jednak z dodatkowymi kosztami.

## Wnioski

1. Aplikacja sleeves.py umożliwia analizę stopnia porażenia poszczególnych liści i usunięcie obserwacji odstających z analizy zmniejszając błąd pomiarowy lub błędy grube w doświadczeniu.
2. Aplikacja umożliwia analizę z wykorzystaniem średniej ważonej stopnia porażenia.
3. Nie stwierdzono istotnie statystycznie różnic pomiędzy porównywanymi metodami analizy objawów chorobowych septorioz.
4. Aplikacja umożliwia przetwarzanie wsadowe.
5. Aplikacja w pełni wykorzystuje moc obliczeniową CPU wielowątkowych.

## Literatura

- Easlson and Bloom, (2014). Easy leaf area: automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in Plant Sciences*. 2 (7): 1400033.
- Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T. and Eliceiri, K. W. (2017). ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics* 18:529.
- Singh, V., Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture* (4) 41–49.
- Thomas, S., Behmann, J., Steier, A., Kraska, T., Muller, O., Rascher, U. and Mahlein, A. K. (2018). Quantitative assessment of disease severity and rating of barley cultivars based on hyperspectral imaging in a non-invasive, automated phenotyping platform. *Plant Methods* 14:45.

