

# Kategoryzacja chorób jabłoni na podstawie wyglądu ich liści

Mateusz Garczyński, Zuzanna Twardowska,  
Wiktor Wołek, Dawid Wysocki



Rysunek 1: Choroby widoczne na liściach jabłoni. Źródło: kaggle [1].

## 1 Opis tematu

Choroby liści jabłoni stanowią jedno z głównych zagrożeń dla produkcji jabłek. Dotychczas jedyny sposób ich wykrywania opiera się na manualnej i czasochłonnej pracy ludzkiej, co często okazuje się zbyt mało efektywne przy ciągle rosnącym popycie. Tym samym narzędzie mogące kategoryzować drzewa na podstawie wyglądu ich liści może przyspieszyć i poprawić wykrywanie chorób w sadach jabłoni.

Jednym z głównych wyzwań dla modeli komputerowych kategoryzujących choroby na podstawie liści jest różnorodność wizualnych symptomów chorób pomiędzy pojedynczymi okazami, a także różnice wynikające z wieku zmian chorobowych, wieku poszczególnych okazów czy warunków w jakich przebiegała uprawa.

Celem projektu jest stworzenie modelu machine-learningowego do dokładnej klasyfikacji chorób jabłoni na podstawie zdjęć ich liści i identyfikacji poszczególnych chorób ze zdjęć liści przedstawiających wiele symptomów.

Dane potrzebne do stworzenia modelu pochodzą z platformy kaggle [1]. Zbiór danych podzielony jest na zestaw treningowy i zestaw testowy. Obydwa zestawy składają się ze zdjęć liści jabłoni opisanych klasami reprezentującymi choroby, których symptomy są widoczne na liściu. W przypadku, gdy pojedynczy liść okazuje symptomy więcej niż jednej choroby opis stanowi listę tych chorób lub zdjęcie opisane jest poprzez klasę "complex" jeśli ilość chorób jest zbyt duża do zidentyfikowania poszczególnych chorób.

## 2 Założenia technologiczne

Poniżej zostaną przedstawione założenia technologiczne projektu.

### 2.1 Język

Projekt zostanie zaimplementowany za pomocą języka Python w wersji 3, który swoją popularność w dziedzinie uczenia maszynowego zawdzięcza dużej ilości bibliotek o takim przeznaczeniu.

### 2.2 Biblioteki

W projekcie zostanie wykorzystana biblioteka TensorFlow i biblioteka Keras pełniąca rolę interfejsu biblioteki TensorFlow, z uwagi na ich dużą popularność w dziedzinie uczenia maszynowego [2]. TensorFlow zostało wydane przez Google w 2015 roku.

W związku z koniecznością przetwarzania obrazów wykorzystana zostanie również biblioteka OpenCV, które jest najpopularniejszą biblioteką do wizji komputerowej [3]. Jest zaimplementowana w C, ale dzięki API może być wykorzystywana w Pythonie. Obecnie posiada ona ponad 2500 algorytmów i oferuje duże możliwości związane z object recognition.

### 2.3 Narzędzia do przetwarzania danych

Do obróbki danych wykorzystamy równie popularne: numpy oraz pandas. NumPy jest biblioteką wręcz wymaganą przy wszelkiego rodzaju zaawansowanych obliczeniach umożliwiającą tworzenie wielowymiarowych tablic i wykonywanie na nich szybkich operacji. Pandas natomiast rozszerza NumPy o funkcje znacznie przyspieszające żmudne i repetytywne zadania takie jak wczytywanie, oczyszczanie i normalizacja danych czy też ich analiza statystyczna.

### 2.4 Prezentacja wyników

Aby zaprezentować dane, użyjemy bibliotek matplotlib oraz seaborn. Matplotlib jest biblioteką umożliwiającą tworzenie zarówno statycznych, animowanych jak i interaktywnych wizualizacji danych w Pythonie. Jest ona bazą dla wielu innych paczek rozbudowujących jej funkcjonalności m.in. dla biblioteki Seaborn. Seaborn jako biblioteka rozbudowująca funkcjonalności Matplotlib umożliwia tworzenie zaawansowanych wykresów, które może okazać się konieczne na etapie prezentacji wyników projektu.

### 2.5 Środowisko treningowe

Przetwarzanie modeli wymaga wykonywania wielu powtarzalnych operacji, w szczególności mnożenia macierzy. Algorytmy implementowane w bibliotekach są często równoległe, dzięki czemu działają znacznie szybciej na kartach graficznych, które w przeciwieństwie do CPU są przystosowane do tego rodzaju pracy.

Jednak ze względu na wielkość danych i mnogość modeli, trenowanie i sprawdzanie skuteczności modeli nie może odbyć się na komputerach osobistych. Do tego zostanie wykorzystana odpowiednia usługa chmurowa, od jednego z dostawców: AWS, Azure, GCP, lub za zgodą wydziału, komputery na serwerze MiNI.

### 3 Podział zadań

Zadania dzielą się na dwie główne kategorie - obróbka danych i trenowanie modeli. W ten sposób postanowiliśmy podzielić się zadaniami. Dawid Wysocki i Zuzanna Twardowska zajmą się przede wszystkim trenowaniem modeli, a Mateusz Garczyński i Wiktor Wólek wstępną i końcową obróbką danych i wyników uzyskanych podczas realizacji projektu. Podział zadań może ulec zmianie w trakcie realizacji projektu.

### Literatura

- [1] Platforma kaggle.com: <https://www.kaggle.com/c/plant-pathology-2021-fgvc8/>
- [2] Keras: <https://keras.io/>
- [3] OpenCV: <https://opencv.org/about/>