

# Rozpoznawanie Liści



<http://yesofcorsa.com/autumn-leaves/>

## *Informatyka Medyczna*

Jan Kosiński i Alicja Lewandowska

Informatyka Medyczna

# Program

Wymagania:

system operacyjny LINUX/MAC OS X

Python3 wraz z pakietami matplotlib, numpy, , open-cv,, scikit-image, scikit-learn

## featureExtraction.py

Pierwszy z trzech skryptów wchodzących w skład narzędzia służy do ekstrakcji cech liści ze zdjęć służących do stworzenia i ewaluacji klasyfikatora. Skrypt jest kompatybilny z Pythonem 3. Pomoc w uruchomieniu można uzyskać poprzez wywołanie polecenia *featureExtraction.py -h*. Do uruchomienia skryptu niezbędne jest użycie parametrów *-i/—input Katalog* oraz *-o/—output plik\_wynikowy.csv*. Katalog powinien zawierać tylko i wyłącznie katalogi reprezentujące poszczególne gatunki liści. W tych katalogach winny znajdować się zdjęcia liści (liść umiejscowiony na środku kadru na jasnym tle). Program ignoruje pliki ukryte (rozpoczynające się od „.”).



Rysunek 1. Przykładowe zdjęcie liścia (*Ginkgo biloba*)

### Cechy na podstawie których budowany jest klasyfikator:

- ◆ obwód konturu
- ◆ postrzępienie blaszki liściowej (kontur jest dopasowywany na dwóch poziomach szczegółowości, mierzony jest stosunek ich długości; wyższa wartość współczynnika oznacza bardziej postrzępione krawędzie; zakres wartości od 0 do 1)

- ◆ stosunek długości convex hull do długości konturu (pozwała zidentyfikować liście składających się z większej liczby podjednostek)
- ◆ stosunek długości i szerokości minimalnego opisującego obiekt prostokąta (rotacja możliwa; parametr świadczy o proporcjach liścia)
- ◆ promień najmniejszego okręgu opisującego liść
- ◆ stosunek pola najmniejszego opisanego na liściu koła do najmniejszego opisanego na liściu prostokąta (parametr świadczący o proporcjach)
- ◆ Parametry zaimplementowane w bibliotece scikit-image: `eccentricity`, `major_axis_length`, `minor_axis_length`, `extent`, `solidity`, `filled-area`, `euler_number`

## leavesRecognition.py

Drugi z trzech skryptów wchodzących w skład narzędzia służy do stworzenia i ewaluacji klasyfikatora. Skrypt jest kompatybilny z Pythonem 3. Pomoc w uruchomieniu można uzyskać poprzez wywołanie polecenia `leavesRecognition.py -h`. Do uruchomienia skryptu niezbędne jest użycie parametrów `-i/—input` ścieżka\_do\_pliku\_wynikowego\_featureExtraction. Przy pomocy opcjonalnego parametru `-s/—sep` można zdefiniować separator pól w pliku csv. Domyślnie jest to znak `\t`.

Do klasyfikacji wykorzystany został algorytm Random Forest z pakietu `scikit-learn`, który charakteryzował się najwyższym % poprawnie zaklasyfikowanych przypadków spośród wszystkich przetestowanych algorytmów.

## leavesClassification.py

Trzeci skrypt służy do rozpoznawania zadanych zdjęć liści przy pomocy utworzonego wcześniej klasyfikatora. Skrypt jest kompatybilny z Pythonem 3. Pomoc w uruchomieniu można uzyskać poprzez wywołanie polecenia `leavesClassification.py -h`. Do uruchomienia skryptu niezbędne jest użycie parametru `-i/—input` oraz podanie ścieżki do katalogu zawierającego zdjęcia liści do rozpoznania. Przy pomocy opcjonalnego parametru `-d/—dir` należy podać ścieżkę do katalogu zawierającego wszystkie skrypty wchodzące w skład narzędzia, jeżeli skrypt nie jest uruchamiany z tego właśnie katalogu.

# Klasyfikator

Jako klasyfikator użyty do rozpoznawania gatunków liści wykorzystane zostały lasy losowe (RandomForestClassifier z biblioteki scikit-learn). Przetestowane zostały również inne klasyfikatory lecz wyniki otrzymywane przy pomocy wszystkich pozostałych metod mocno odstawały od wybranej.

Do wyboru cech użytych do klasyfikacji posłużyła funkcja SelectKBest (statystyka chi2) z biblioteki scikit-learn uruchomiona dla każdej możliwej liczby całkowitej z zakresu [1:12]. Najlepszy klasyfikator otrzymano przy użyciu wszystkich cech jako zbioru uczącego.

Liczba cech	1	2	3	4	5	6	7	8	9	10	11	12
Jakość klasyfikacji	0.20	0.39	0.53	0.56	0.56	0.56	0.66	0.65	0.66	0.77	0.77	0.78

Tabela 1. Jakość klasyfikacji dla nauczania klasyfikatora na bazie określonej liczby najlepszych cech wg funkcji SelectKBest dla przykładowego uruchomienia algorytmu.

## Otrzymany klasyfikator:

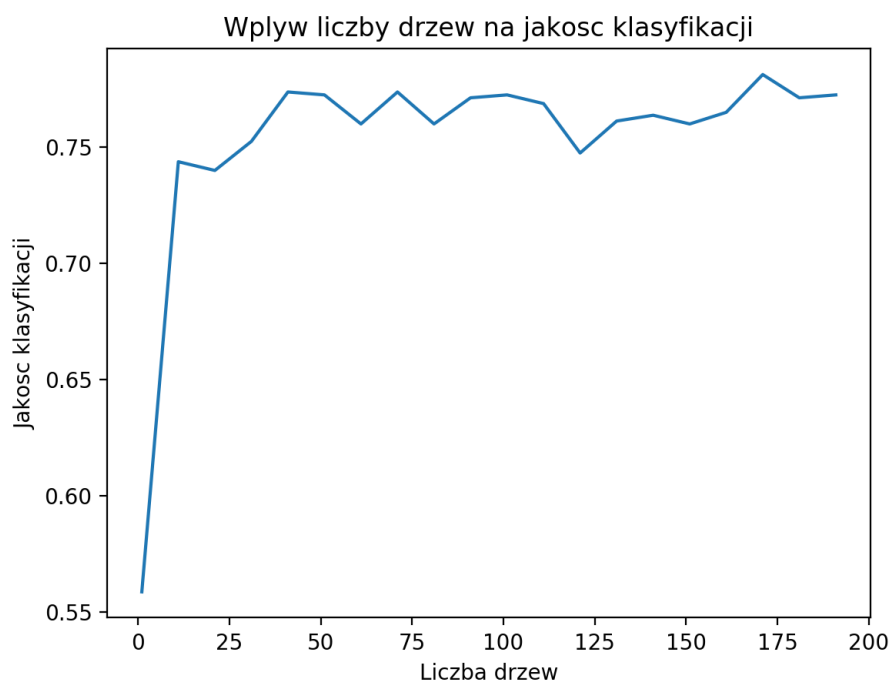
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',  
                        max_depth=10, max_features='auto', max_leaf_nodes=None,  
                        min_impurity_split=1e-07, min_samples_leaf=1,  
                        min_samples_split=2, min_weight_fraction_leaf=0.0,  
                        n_estimators=171, n_jobs=1, oob_score=False, random_state=None,  
                        verbose=0, warm_start=False)
```

```
{'n_estimators': 171, 'max_depth': 10, 'criterion': 'gini'}
```

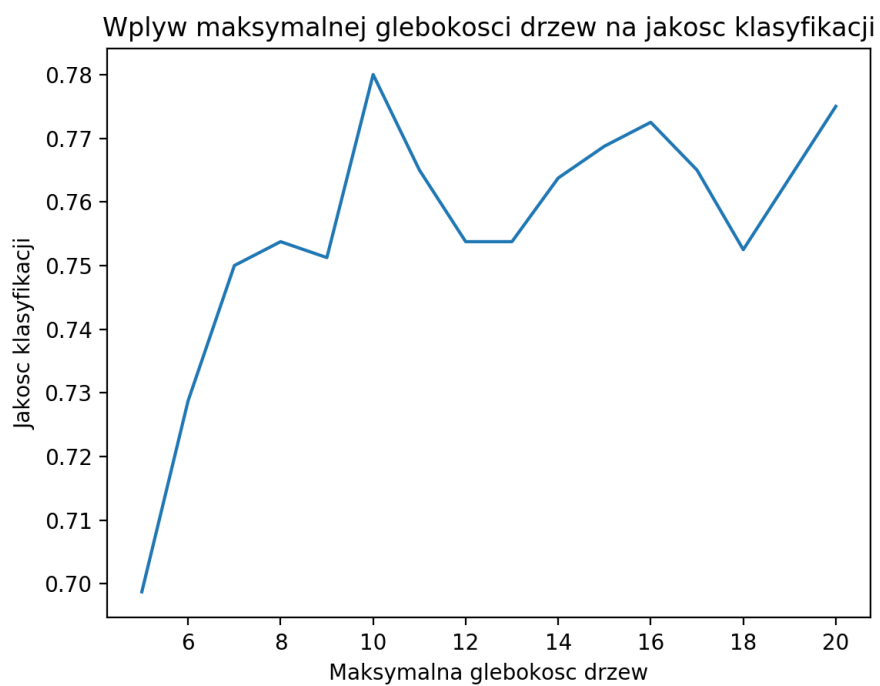
**Jakość klasyfikacji:** 0.78375

## Wpływ poszczególnych parametrów na jakość klasyfikacji

Ze względu na wysoką złożoność czasową funkcji testującej poszczególne parametry modelu przy użyciu 10-krotnej walidacji krzyżowej, w szczegółowy sposób zostały przetestowane kluczowe parametry klasyfikatora, tj. n\_estimators (liczba drzew w lesie), criterion (funkcja mierząca jakość podziału) oraz max\_depth (maksymalna głębokość drzew w lesie).



Wykres 1. Wpływ liczby drzew w lesie na jakość otrzymanego klasyfikatora.



Wykres 2. Wpływ maksymalnej głębokości drzewa w lesie na jakość otrzymanego klasyfikatora.