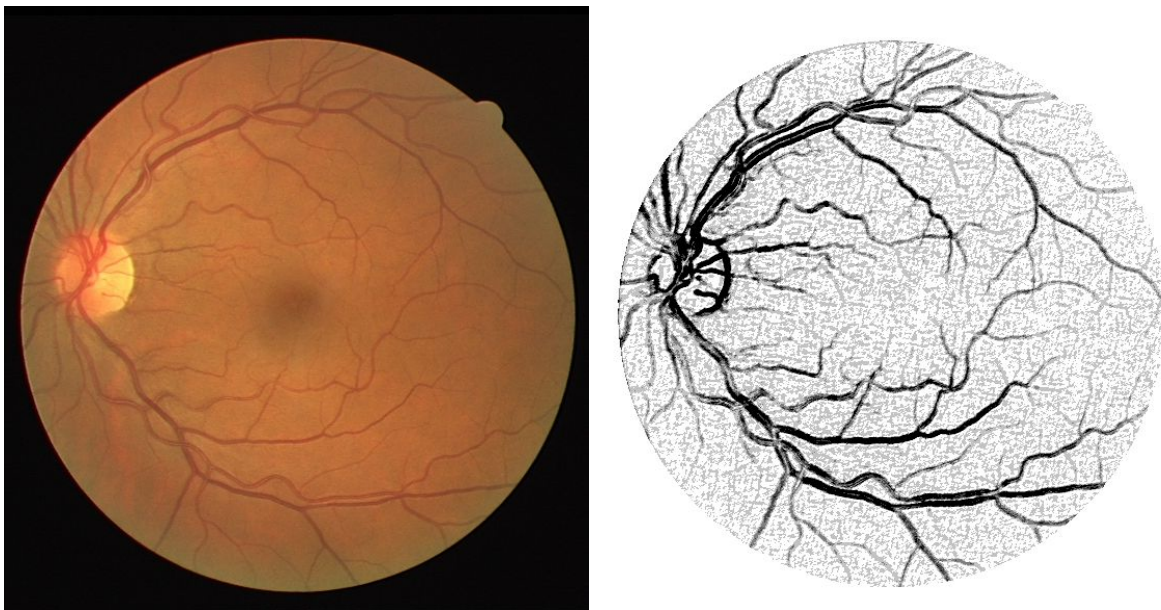


# Wykrywanie naczyń dna siatkówki oka

## Opis

Przygotowanie aplikacji (okienkowa, Jupyter notebook, ...), która dla zadanego obrazu wejściowego przedstawiającego dno siatkówki oka (przykład poniżej) automatycznie wykrywa naczynia krwionośne. Z formalnego punktu widzenia dla każdego piksela wykorzystany algorytm musi stwierdzić, czy ten piksel stanowi naczynie krwionośne, czy nie (klasyfikacja binarna).



## Wymagania obowiązkowe

- Algorytm w podstawowej wersji powinien wykorzystywać techniki przetwarzania obrazu (poznane między innymi na przedmiocie KCK – zadanie z samolotami/projekt z obrazów) do detekcji naczyń krwionośnych. W ramach takiego procesu przetwarzania można wyróżnić 3 główne elementy:
  - a) *Wstępne przetworzenie obrazu*: wejściowy obraz może być zaszumiony/zbyt ciemny/jasny. Można tutaj wykorzystać takie techniki jak: rozmycie, wyostrenie, normalizacja histogramu kolorów itp.
  - b) *Właściwe przetworzenie obrazu*: w celu wyodrębnienia naczyń krwionośnych: można zastosować różne techniki wykrywania krawędzi (np. filtr Frangi’ego).
  - c) *Końcowe przetwarzanie obrazu*: przetwarzanie uzyskanego obrazu w celu poprawy skuteczności wykrywania naczyń poprzez “naprawę” błędów z poprzednich kroków.

- Wynik obowiązkowo należy wizualizować np. zamalowując wyróżniającym się kolorem piksele zaklasyfikowane jako naczynie krwionośne. W tym celu najlepiej wygenerować binarną maskę odpowiedzi algorytmu, która zostanie potem wykorzystana do analizy statystycznej (porównanie z maską ekspercką z ręcznie zaznaczonymi naczyniami).
- Ważnym elementem oceny jest skuteczność algorytmu. W tym celu należy dokonać podstawowej analizy statystycznej jakości działania algorytmu. Działanie programu należy przetestować na minimum 5 obrazach. Podczas testów należy wyznaczyć macierze pomyłek oraz takie miary jak *trafność (accuracy)*, *czułość (sensitivity)*, *swoistość (specificity)*. Przy wyznaczaniu czułości i swoistości należy założyć, że naczynie to klasa pozytywna, a tło - negatywna. Ponieważ mamy do czynienia z niezrównoważonym rozkładem klas (punktów tła jest znacznie więcej niż naczyń) należy dodatkowo wykorzystać miary dla danych niezrównoważonych (np. średnią arytmetyczną lub geometryczną czułości i swoistości).

## Wymagania na 4.0

- Po wstępnym przetworzeniu obrazu należy podzielić go na wycinki (np. 5x5 px) i dla każdej z nich dokonać ekstrakcji cech z obrazu: np. wariancja kolorów, momenty centralne, momenty Hu itp. Wartości te wraz z informacją pochodzącą z maski eksperckiej (decyzja dla środkowego piksela wycinka) stanowią zbiór danych wykorzystany do budowy prostego klasyfikatora odległościowego (np. kNN). Można skorzystać z gotowej implementacji klasyfikatora (np. w scikit-learn).
- Z uwagi na ograniczenia pamięciowe konieczne może być ograniczenie rozmiaru zbioru uczącego poprzez losowy wybór punktów (możliwość zastosowania *undersampling-u* do zrównoważenia rozkładu klas w zbiorze uczącym).
- Zdolności predykcyjne tak opracowanego klasyfikatora należy wstępnie zweryfikować na niezależnym zbiorze testowym *hold-out* (np. pochodzącym z innej części obrazu lub z innego obrazu).
- Gotowy klasyfikator powinien zostać osadzony w aplikacji, a jego działanie powinno zostać zwizualizowane i przetestowane w taki sam sposób, jak działanie technik przetwarzania obrazu z wymagań podstawowych..

## Wymagania na 5.0

- Przygotowanie danych takie samo, jak w przypadku wymagań na 4.0. Należy jednak wykorzystać bardziej zaawansowany klasyfikator, np.: drzewo decyzyjne lub las, reguły decyzyjne, sieć neuronową. Można wykorzystać gotowe implementacje klasyfikatorów (scikit-learn, WEKA, ...).

- Należy wykorzystać k-krotną walidację skrośną (*k-fold cross validation*) w celu wstępnej oceny zbudowanego klasyfikatora i uniknięcia przeuczenia.
- Zbudowany klasyfikator powinien zostać osadzony w aplikacji i tam dodatkowo przetestowany zgodnie z wymaganiami obowiązkowymi.

## Uwaga

Realizując wymagania na 4.0 lub 5.0 należy także zrealizować wymagania obowiązkowe -- wyniki uzyskane za pomocą prostych metod filtrowania obrazu będą stanowić punkt odniesienia (baseline) dla bardziej zaawansowanych modeli decyzyjnych. Realizując wymagania na 5.0 należy pominąć wymagania na 4.0 (nie ma konieczności budowy dwóch klasyfikatorów - mniej i bardziej złożonego).

W projekcie należy skorzystać z jednej z dostępnych baz danych z obrazami (patrz linki poniżej) -- ta sama baza powinna być stosowana we wszystkich krokach projektu.

## Raport

Raport powinien zawierać następujące elementy (część z nich dotyczy wymagań na 4.0 i 5.0 - zostało to zaznaczone):

1. Skład grupy
2. Zastosowany język programowania oraz dodatkowe biblioteki
3. Opis zastosowanych metod:
  - a. przetwarzanie obrazów
    - i. poszczególne kroki przetwarzania obrazu (w tym zastosowane filtry),
    - ii. krótkie uzasadnienie zastosowanego rozwiązania,
  - b. uczenie maszynowe (4.0/5.0)
    - i. przygotowanie danych - wyznaczanie wycinków obrazu, ekstrakcja cech z wycinków
    - ii. wstępne przetwarzanie zbioru uczącego (jeśli zastosowano)
    - iii. zastosowane metody uczenia maszynowego wraz z informacją o przyjętych parametrach
    - iv. wyniki wstępnej oceny zbudowanego klasyfikatora (testy *hold-out* lub *k-fold cross validation*)
    - v. krótkie uzasadnienie zastosowanego rozwiązania
4. Wizualizacja wyników działania programu dla wybranych obrazów (warto pokazać zarówno sukcesy jak i porażki). Dla porównania należy zamieścić maskę ekspercką (*ground truth*). Wyniki wizualizacji należy zaprezentować dla co najmniej 5 obrazów (w przypadku metod uczenia maszynowego nie mogą to być obrazy, które zostały wykorzystane do nauczania klasyfikatora).

5. Analiza wyników działania programu dla wybranych obrazów (tych samych, które wykorzystano w punkcie 4) z wykorzystaniem odpowiednich miar oceny (omawianych wcześniej). Analizę należy przeprowadzić indywidualnie dla każdego z obrazów. W przypadku realizacji zadań na 4.0/5.0 należy dokonać porównania miar oceny osiągniętych przez metody przetwarzania obrazów oraz uczenia maszynowego.

## Linki

- Baza obrazów HRF: <https://www5.cs.fau.de/research/data/fundus-images/> (proszę się nie przejmować komunikatem “website deprecated and outdated”).
- Baza obrazów STARE: <http://cecas.clemson.edu/~ahoover/stare/probing/>
- Baza obrazów CHASE: <https://blogs.kingston.ac.uk/retinal/chasedb1/> (do przeglądania zdjęć wymagany jest niestety Flash, dlatego najlepiej od razu pobrać całe archiwum)