**Informatyka w medycynie – siatkówka oka**

1. **Język programowania:**  
   -python

**biblioteki:**

* matplotlib
* cv2
* tabulate
* numpy
* sklearn
* imblearn
* sklearnex
* skiimage

1. Zastosowane metody:

**Przetwarzanie obrazów**

**Wczytanie obrazu i konwersja przestrzeni barw**

* + Obraz jest wczytywany za pomocą cv2.imread i konwertowany z BGR do RGB
  + Wyświetlane są poszczególne kanały (R, G, B) oraz obraz w skali szarości
  + Wykorzystanie kanału zielonego (G) jako podstawy dalszej analizy, ponieważ naczynia krwionośne są najbardziej widoczne w tym kanale

**Zastosowanie CLAHE – wyrównanie histogramu (Contrast Limited Adaptive Histogram Equalization),który ma na celu poprawę kontrastu obrazu**

* + Wybór kanału zielonego i ograniczenie wartości pikseli do zakresu 10-245
  + Zastosowanie CLAHE z parametrami clipLimit=2.0(próg ograniczenia kontrastu) i tileGridSize=(8, 8) (8×8=64 kafelki)
  + CLAHE poprawia lokalny kontrast obrazu, co jest kluczowe dla uwydatnienia naczyń krwionośnych przy zachowaniu szczegółów

**Zastosowanie filtru Frangiego**

* + Filtr Frangiego jest stosowany z parametrem sigmas=np.arange(1, 5, 0.5)(od 1 do 5 z korkiem co pół) i black\_ridges=True(wykrywa ciemne grzbiety na białym tle)
  + Normalizacja wyniku i progowanie z wartością 0.065(img\_frangi > 0.065, 1, 0)
  + Usunięcie obramowania obrazu za pomocą funkcji remove\_border
  + Filtr Frangiego jest specjalnie zaprojektowany do wykrywania struktur rurowych (jak naczynia krwionośne) poprzez analizę wartości własnych macierzy Hessego

**Sekwencyjne operacje morfologiczne otwarcia i zamknięcia**

* Zastosowanie elementów strukturalnych w kształcie elipsy o różnych rozmiarach (5, 7, 15, 23)
  + Operacje otwarcia usuwają małe obiekty i szumy
  + Operacje zamknięcia wypełniają małe dziury i łączą bliskie struktury
  + Sekwencyjne zwiększanie rozmiaru jądra pozwala na stopniowe usuwanie coraz większych struktur

**Usuwanie tła**

Usunięcie tła z obrazu

* + Wykorzystanie ostatniego obrazu z sekwencji operacji morfologicznych jako przybliżenia tła
  + Odjęcie tła od oryginalnego obrazu w skali szarości za pomocą cv2.subtract
  + Zastosowanie rozmycia Gaussa (5x5) do wygładzenia wyniku
  + Aby uczynić naczynia jeszcze bardziej widocznymi, zdecydowaliśmy się zastosować rozmycie Gaussa w celu usunięcia szumów z obrazu oryginalnego

**Normalizacja obrazu po usunięciu tła**

* + Ograniczenie wartości pikseli do zakresu 0-20
  + Normalizacja do pełnego zakresu 0-255
  + Zastosowanie CLAHE dla poprawy kontrastu do obrazu po usunięciu tła
  + Normalizacja poprawia kontrast i ułatwia segmentację naczyń

**Binaryzacja i porównanie z maską eksperta**

* + Dla celów porównania obrazów, przyjmujemy, że (w skali od 0 do 1), piksele o wartości powyżej 0,75 otrzymują wartość 1, natomiast piksele o wartościach poniżej tej granicy są traktowane jako 0. Jest to binarna maska odpowiedzi algorytmu.

**Ewaluacja wyników**

Obliczenie metryk oceny jakości segmentacji

* + Obliczenie macierzy pomyłek
  + Wyliczenie metryk:
    1. dokładność (accuracy) oznacza, że X% wszystkich pikseli zostało poprawnie sklasyfikowanych
    2. czułość (sensitivity)  oznacza, że X% pikseli naczyń zostało poprawnie wykrytych
    3. swoistość (specificity) oznacza, że X% pikseli tła zostało poprawnie zidentyfikowanych jako tło
    4. zbalansowana dokładność, to średnia arytmetyczna czułości i swoistości
    5. średnia geometryczna, to pierwiastek kwadratowy z iloczynu czułości i swoistości
  + Metryki te są kluczowe do obiektywnej oceny jakości segmentacji, szczególnie w przypadku niezbalansowanych danych (gdzie piksele naczyń stanowią mniejszość)

**Uczenie maszynowe**

W procesie przygotowania danych zastosowano metodę wycinków (patch-based approach), gdzie z każdego obrazu siatkówki pobierano fragmenty o wymiarach 5×5 pikseli. Dla każdego wycinka przeprowadzono ekstrakcję następujących cech:

1. Podstawowe statystyki:
   * Wariancja pikseli w wycinku
   * Średnia wartość pikseli
   * Odchylenie standardowe
2. Wartości ekstremalne:
   * Minimalna wartość piksela
   * Maksymalna wartość piksela
   * Zakres (różnica między wartością maksymalną a minimalną)
3. Momenty obrazu:
   * Momenty Hu (7 wartości)
   * Momenty centralne (mu20, mu11, mu02, mu30, mu21, mu12, mu03)
4. mu20, mu02: Opisują rozproszenie (wariancję) obiektu wzdłuż osi x i y. Związane są z orientacją obiektu.
5. mu11: Moment mieszany, który wskazuje na korelację między wartościami x i y, dostarczając informacji o orientacji obiektu.
6. mu30, mu03: Momenty trzeciego rzędu, które opisują skośność (asymetrię) wzdłuż osi x i y.
7. mu21, mu12: Mieszane momenty trzeciego rzędu, które dostarczają dodatkowych informacji o złożoności kształtu obiektu.

Ekstrakcja cech została zaimplementowana w funkcji extract\_features(), która przekształca każdy wycinek 5×5 w wektor cech o stałej długości. Etykiety dla każdego wycinka określano na podstawie wartości centralnego piksela w masce eksperta (wartość >128 klasyfikowana jako naczynie krwionośne).

**Wstępne przetwarzanie zbioru uczącego**

Przed przystąpieniem do uczenia modelu, przeprowadzono następujące kroki przetwarzania wstępnego:

1. Przetwarzanie obrazów wejściowych:
   * Wykorzystanie kanału zielonego jako najbardziej informatywnego dla struktur naczyniowych
   * Ograniczenie wartości pikseli do zakresu 10-245
   * Zastosowanie rozmycia Gaussa (5×5) w celu redukcji szumów
   * Normalizacja kontrastu za pomocą CLAHE
2. Balansowanie klas:
   * Zastosowanie techniki **Random Undersampling** – w celu zrównoważenia liczby próbek naczyń i tła

rus = RandomUnderSampler(sampling\_strategy=1, random\_state=42) oznacza, że stosunek będzie wynosił 1:1

* + Redukcja zbioru uczącego do maksymalnie 30000 próbek w celu optymalizacji czasu uczenia

1. Podział danych:
   * Wykorzystano 10 pierwszych obrazów jako zbiór treningowy
   * Zastosowano podział train\_test\_split z parametrem test\_size=0.3 do wydzielenia zbioru walidacyjnego

W projekcie zastosowano klasyfikator **Random Forest** z optymalizacją hiperparametrów za pomocą przeszukiwania siatki (Grid Search). Przeszukiwanie przeprowadzono dla następujących parametrów:

* n\_estimators:  - liczba drzew w lesie
* max\_depth:- maksymalna głębokość drzewa
* min\_samples\_leaf: - minimalna liczba próbek wymagana w liściu
* min\_samples\_split: - minimalna liczba próbek wymagana do podziału węzła

Optymalizacja została przeprowadzona z wykorzystaniem **3-krotnej walidacji krzyżowej** (cv=3):

1. W każdej iteracji, 2 z 3 części danych są używane do trenowania modelu
2. Pozostała 1 część służy jako zbiór walidacyjny do oceny wydajności modelu
3. Proces jest powtarzany 3 razy, tak aby każda z 3 części danych raz posłużyła jako zbiór walidacyjny

Cały proces uczenia został zaimplementowany w postaci potoku (pipeline) z wykorzystaniem biblioteki imblearn, co pozwoliło na zintegrowanie procesu balansowania klas z procesem uczenia. Wyniki wychodzą podobne przy zastosowaniu scalera StandardScaler(), dlatego zdecydowaliśmy się na pominięcie tego kroku.

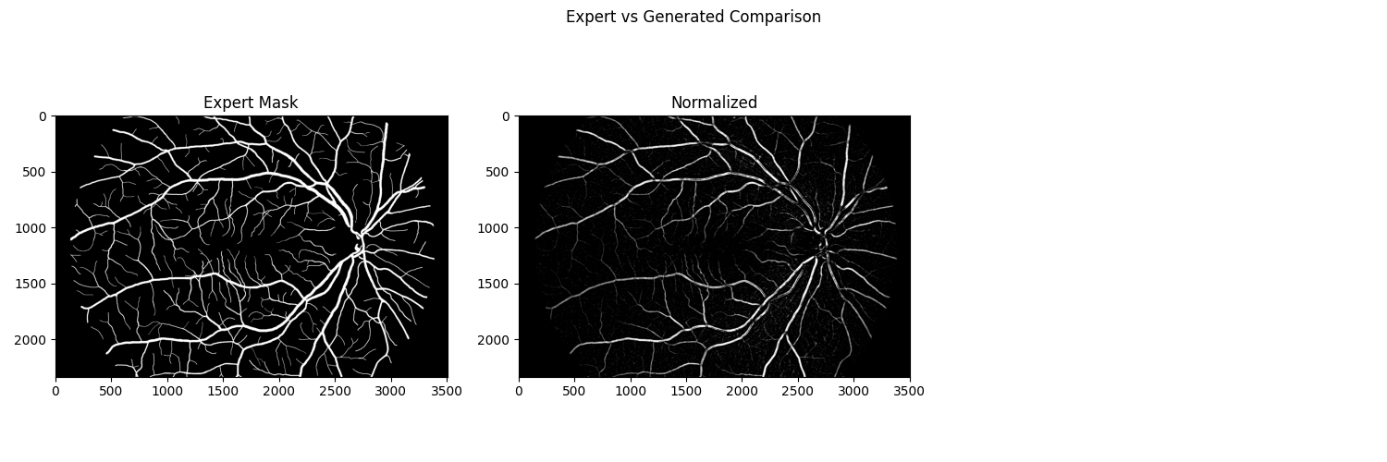
Aby przeprowadzić proces dostrajania modelu, tworzymy obiekt GridSearchCV, który będzie przeszukiwać siatkę parametrów w celu znalezienia najlepszego zestawu dla modelu. Parametr n\_jobs=-1 pozwala na wykorzystanie wszystkich dostępnych rdzeni procesora, przyspieszając tym samym proces dostrajania.

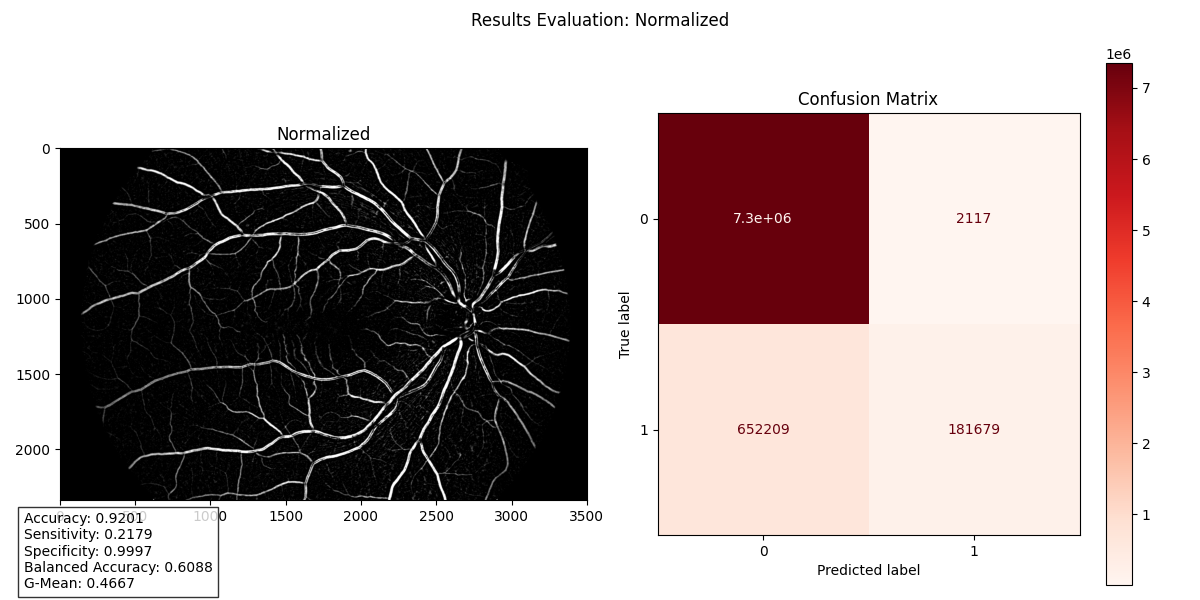
Zastosowane rozwiązanie opiera się na podejściu **patch-based z klasyfikatorem Random Forest**, co jest uzasadnione z następujących powodów:

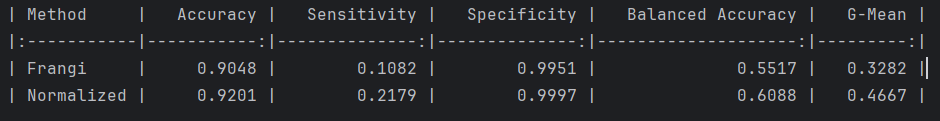
1. Efektywność obliczeniowa - Random Forest oferuje dobrą wydajność przy relatywnie niskich kosztach obliczeniowych w porównaniu do głębokich sieci neuronowych
2. Odporność na przeuczenie - Random Forest jest mniej podatny na przeuczenie dzięki agregacji wielu drzew decyzyjnych, co jest istotne przy ograniczonej liczbie obrazów treningowych
3. Skuteczność przy danych niezbalansowanych - Po zastosowaniu technik balansowania, Random Forest dobrze radzi sobie z problemem niezbalansowanych klas, co jest typowe dla segmentacji naczyń siatkówki (gdzie piksele naczyń stanowią mniejszość).
4. Zastosowanie kanału zielonego oraz technik przetwarzania wstępnego (CLAHE, rozmycie Gaussa) jest zgodne z najlepszymi praktykami w przetwarzaniu obrazów siatkówki, gdzie naczynia krwionośne są najlepiej widoczne właśnie w tym kanale.

Wyniki testów hold-out potwierdzają skuteczność zastosowanego podejścia, szczególnie w kontekście wysokiej swoistości, co jest istotne w zastosowaniach medycznych, gdzie fałszywe pozytywne wyniki mogą prowadzić do błędnych diagnoz. Analizując uzyskane wyniki, można stwierdzić, że istnieje potencjał do dalszej poprawy. Jednym z czynników, który mógłby wpłynąć na lepsze rezultaty, jest zwiększenie liczby próbek tła. Chociaż RandomUnderSampler pozwala na zrównoważenie klas, jego parametry ograniczają się do zakresu 0-1, co może wpłynąć na ogólną jakość modelu.

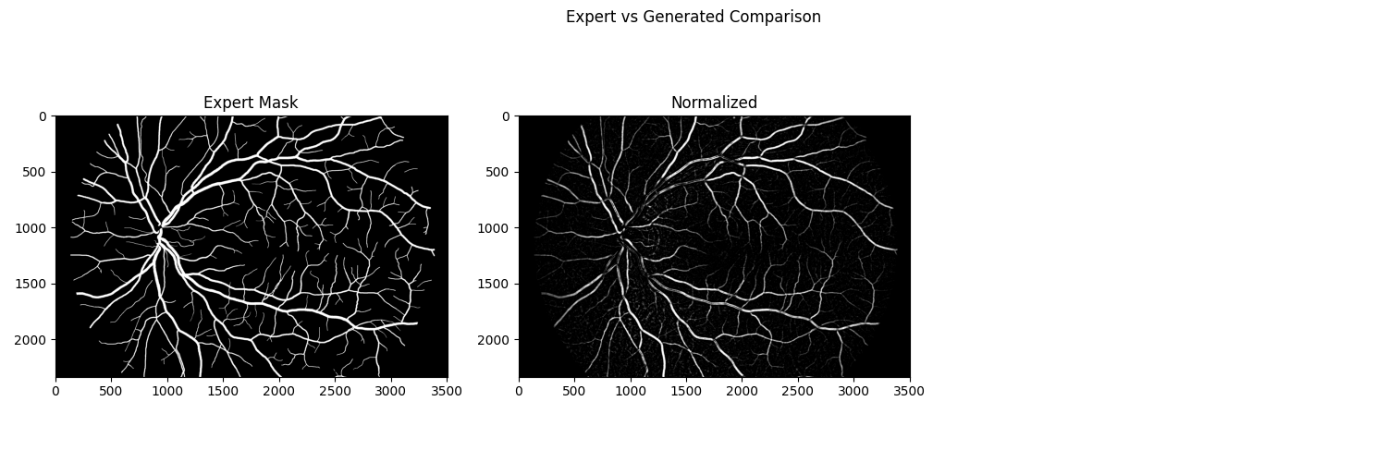
Obraz 1:

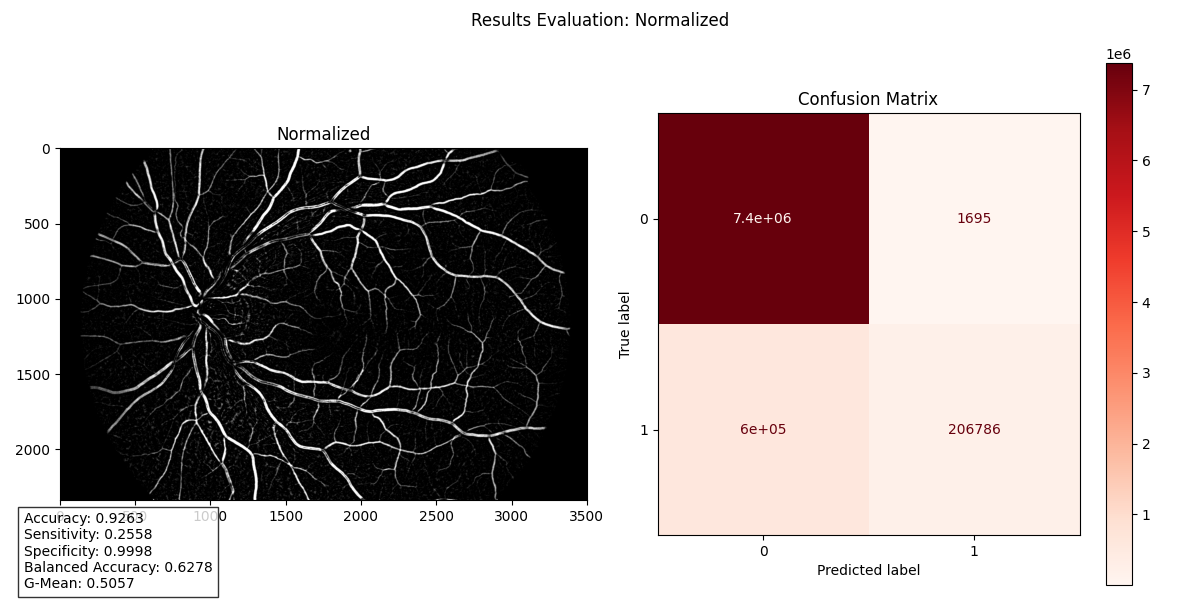


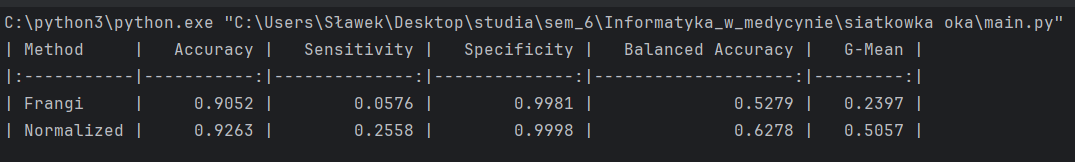




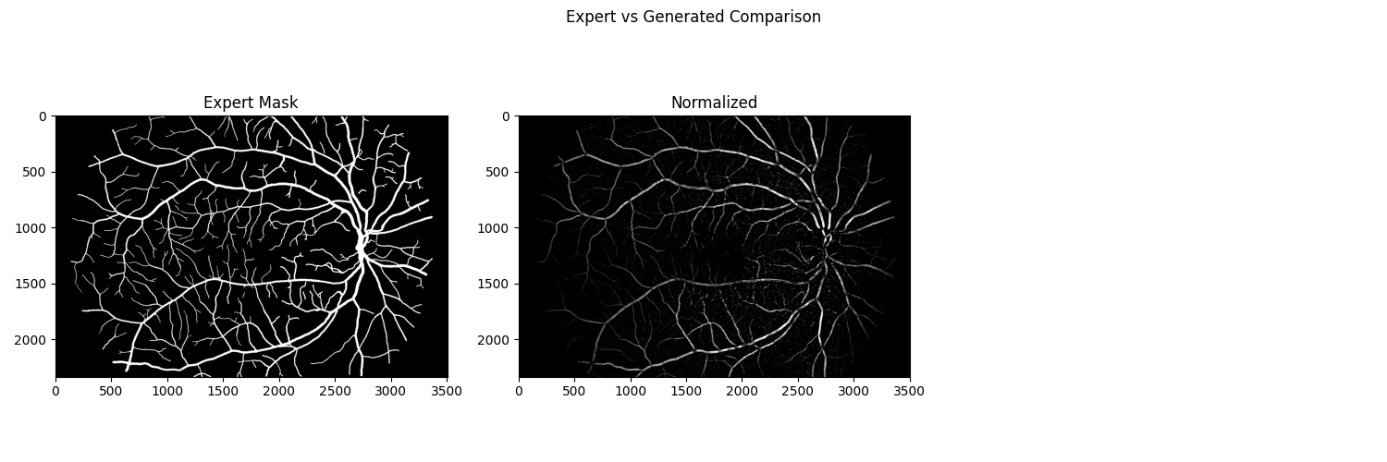
Obraz 2:

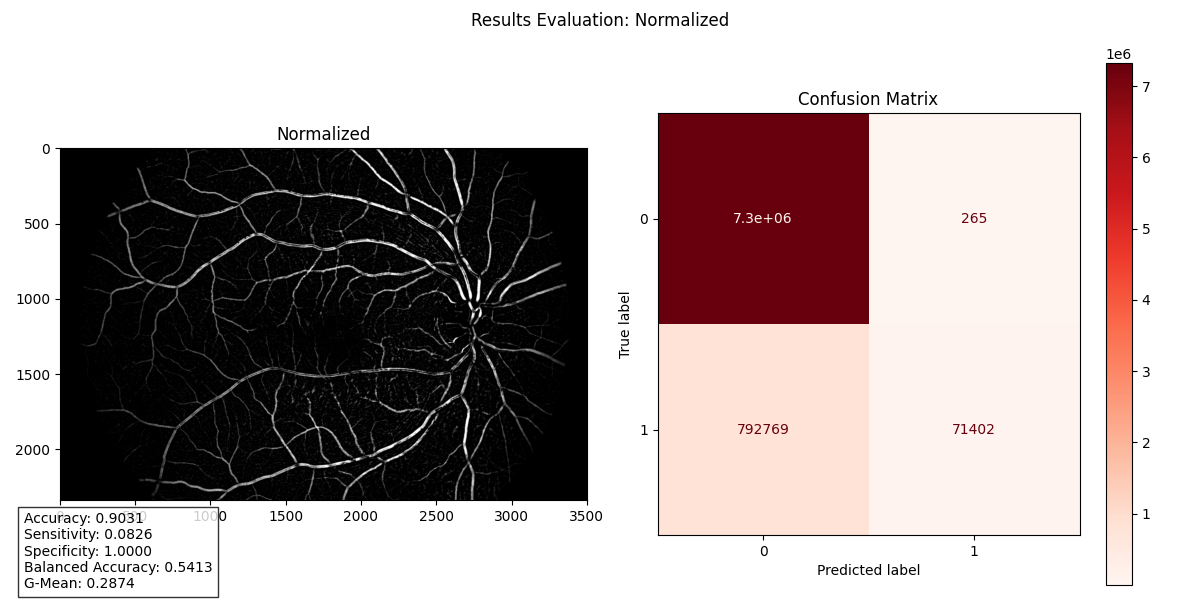


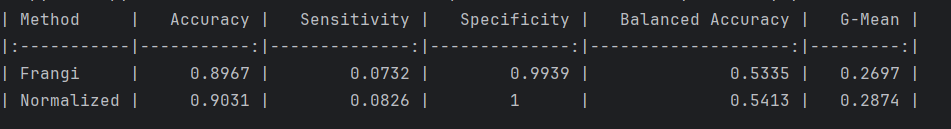




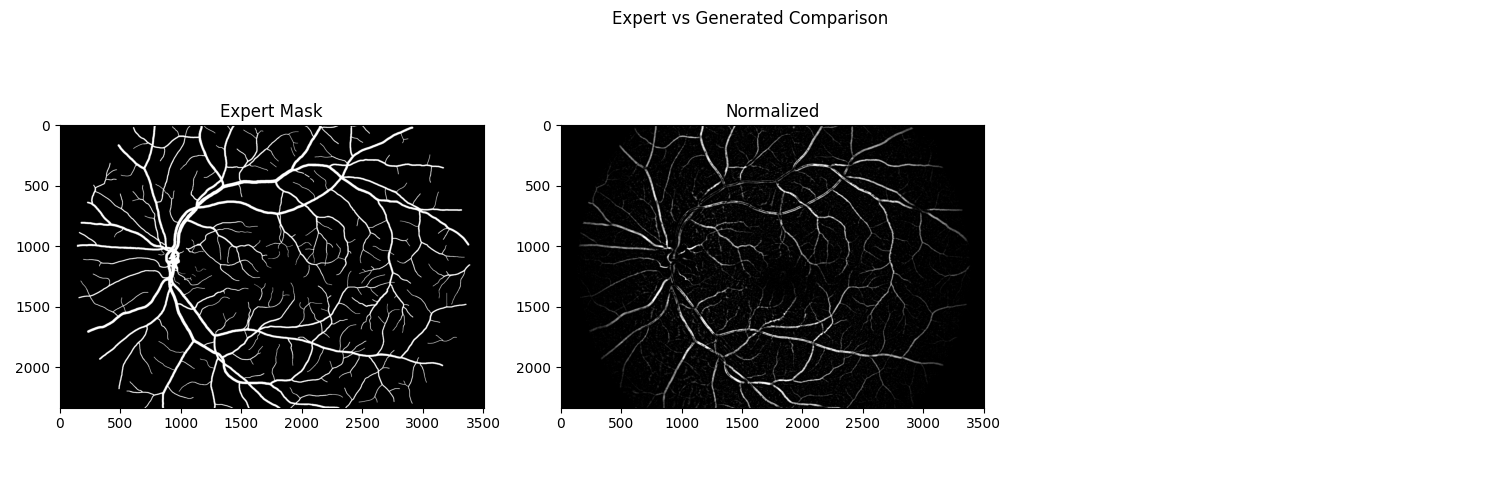
Obraz 3:

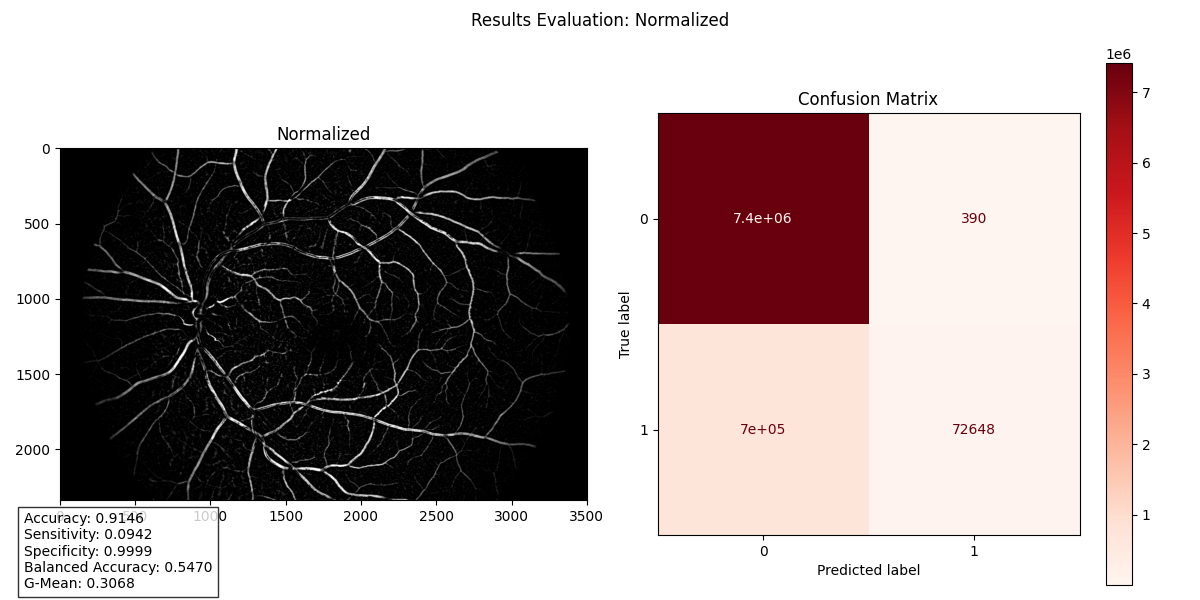


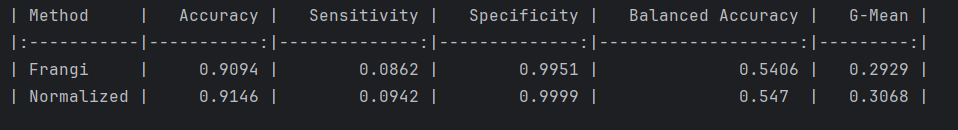




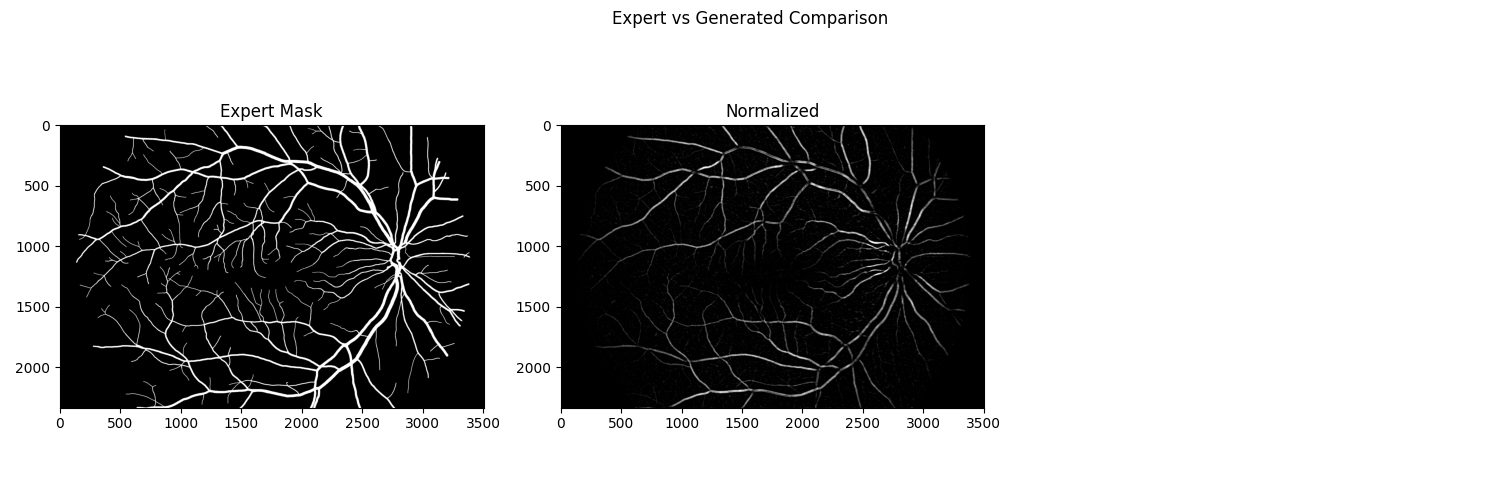
Obraz 4:

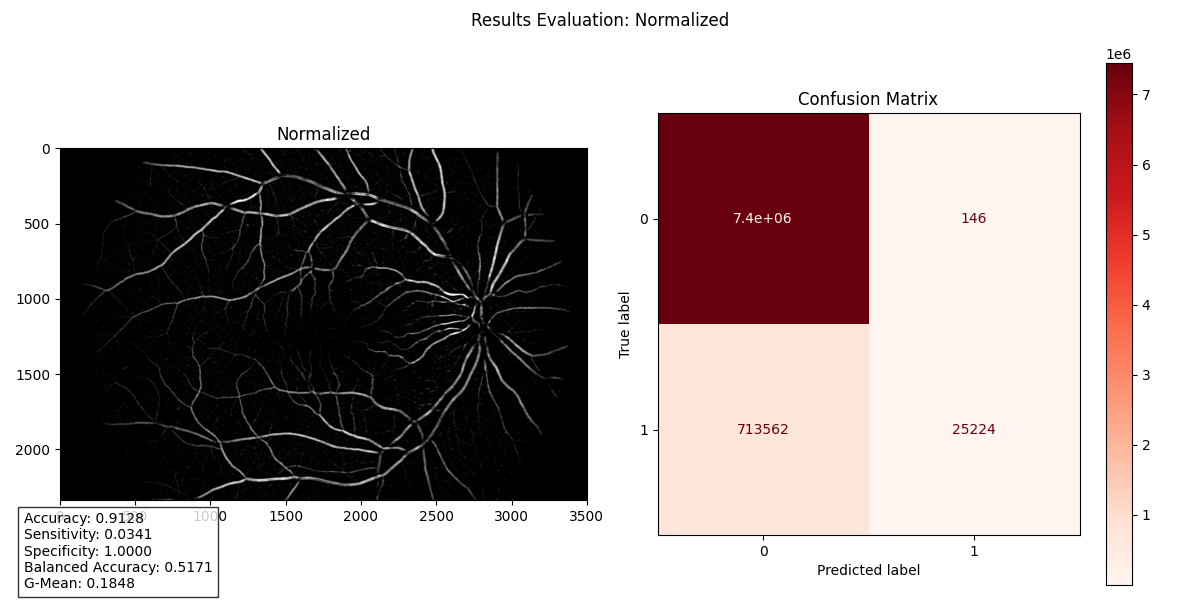


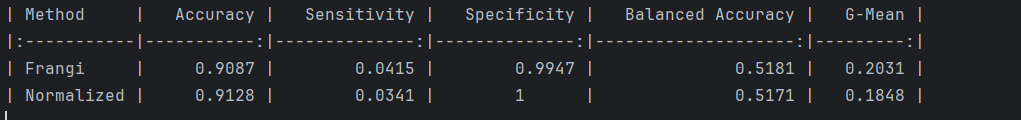




Obraz 5:







Uczenie maszynowe: 