

# Sprawozdanie – retina vessel segmentation

---

## 1. Skład grupy

Stanisław Głowczewski 151877

Rafał Szubert 151792

## 2. Zastosowany język programowania oraz dodatkowe biblioteki

Program zaimplementowano w języku *python3* wraz ze standardową biblioteką *tkinter* do stworzenia GUI. Wykorzystano bibliotekę *numpy* do wspomagania obliczeń, *PIL* oraz *CV2* w celu obróbki obrazów, *matplotlib* do wizualizacji oraz *sklearn*, i *tenserflow* do stworzenia modeli uczenia maszynowego.

## 3. Opis zastosowanych metod:

Wszystkie metody zostały zaimplementowane jako osobne i niezależne moduły. Moduł odpowiedzialny za interfejs użytkownika importuje te moduły oraz wywołuje odpowiednie funkcje (przekazując jako argument przetwarzany obraz). Moduły uczenia maszynowego wymagają wcześniejszego przeszkolenia, odbywa się to poprzez wywołanie głównych metod w odpowiednich plikach.

### a. przetwarzanie obrazów (3.0)

Początkowo zdjęcie poddano wstępnej obróbce. Usunięto kanały czerwony i niebieski oraz wyostrzono obraz. Następnie został nałożony filtr ze zmienną wartością graniczną. Na koniec wykorzystano erozję i dylatację.

Zauważono, że wykorzystanie jedynie zielonego koloru w przetwarzaniu obrazu daje co najmniej tak dobre wyniki, jak połączenie kilku albo wykorzystanie innego koloru. Rodzaj filtra, wartość graniczna oraz wielkość filtra zostały wybrane doświadczalnie poprzez wykonanie wielu testów z różnymi wartościami powyższych parametrów. Otrzymany w ten sposób obraz składał się z wielu kropek- w celu zwiększenia dokładności predykcji użyto połączenia erozji oraz dylatacji.

### b. uczenie maszynowe (4.0)

Przygotowanie modelu składa się z 3 etapów – przygotowania, uczenia oraz ewaluacji. Na początku przygotowujemy dane, poprzez otwarcie zbioru obrazów treningowych, podzielenie ich na odpowiednią ilość mniejszych obrazków (o wymiarach 5x5).

Dla każdego obrazka liczone są miary: *value*, *variance*, *minimum*, *maximum*. Dla każdego obrazka generowany jest również binarny opis mówiący, czy środek danego obrazka jest naczynkiem. Dokonywany jest podział obrazów na zestaw treningowy i testowy.

Zgodnie z założeniami algorytmu „na 4” wykorzystano metodę KNN-*k-nearest neighbors*, która tworzy n-wymiarową przestrzeń ze znanymi z procesu uczenia punktami. W procesie ewaluacji algorytm znajduje k najbliższych sąsiadów wielowymiarowej przestrzeni i przyjmuje wartość najczęściej pojawiającego się stanu

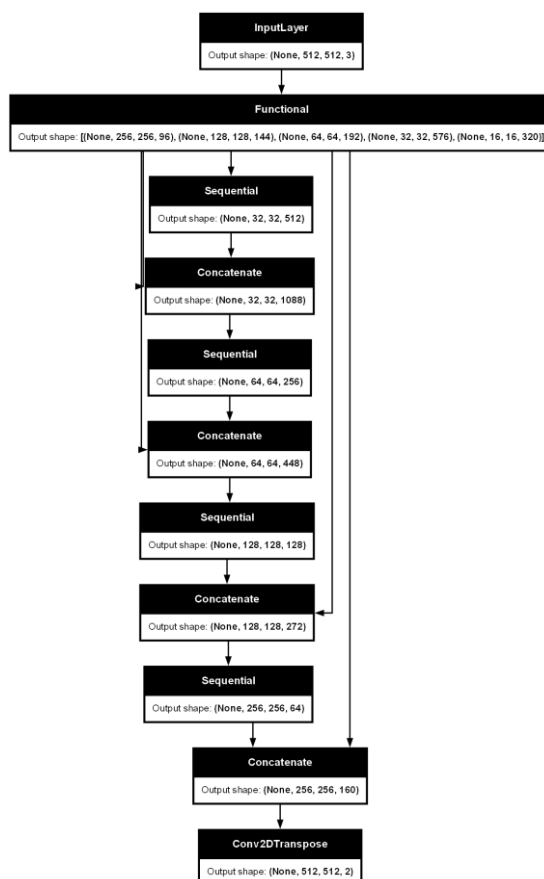
### c. uczenie maszynowe (5.0)

Trenowanie modelu odbywa się w kilku etapach. Na początku wczytywane są obrazy treningowe przy użyciu biblioteki *CV2*. Następnie obrazy są normalizowane oraz zmieniany jest ich wymiar na 512x512 pikseli. W tym celu wykorzystane metody biblioteki *tensorflow*. Obrazy są również losowo obracane w celu zwiększenia dokładności uczenia oraz uodpornienia modelu na przeuczenie cech zależnych od rotacji oka. W kolejnym etapie zbiór dzielony jest na podzbiory treningowy oraz testowy (zbiór treningowy stanowi 70% całego zbioru).

Wykorzystano model klasy U-Net. Jako kodera użyto wstępnie wytrenowanego modelu *MobileNetV2*. Jako dekodery przyjęto serię bloków upsamplingu (zaimplementowanych w przykładach biblioteki *tensorflow*). Procesowi szkolenia poddany został jedynie dekodery. Użyto techniki *skip connections* w celu konkatencji odpowiednich warstw z kodera i dekodera. Do szkolenia modelu użyto optymalizatora *Adam*. Ostatnia warstwa modelu tworzy mapę (obraz) z dwoma kanałami, gdyż rozpoznawanie naczynek jest klasyfikacją binarną. Strukturę modelu zamieszczono na poniższym obrazie.

Wstępne testy *hold-out* wykazały dokładność modelu na poziomie 91.87% (z parametrem *loss* równym 19.81%). Zważywszy na niewielki zbiór treningowy, takie wyniki uznać można za satysfakcjonujące.

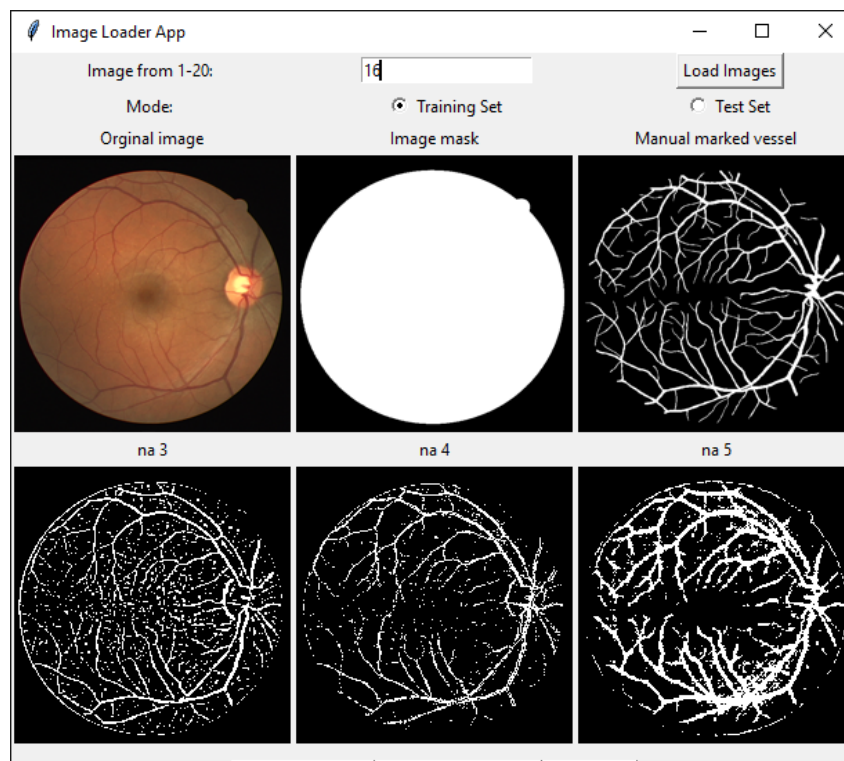
Do realizacji zadania użyto popularnego modelu klasy U-Net, bazującego na konwolucyjnej sieci neuronowej. Ze względu na przystosowanie architektury sieci do precyzyjnego klasyfikowania obrazów oraz trenowania na małych zbiorach, sieci te uzyskują bardzo dobre rezultaty, w względnie krótkim czasie.



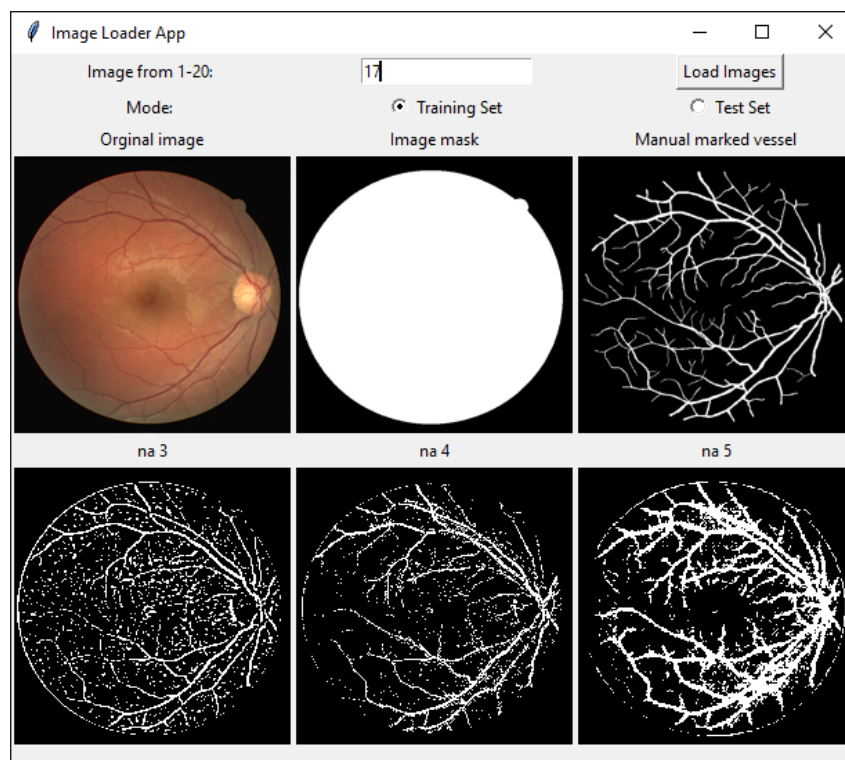
Architektura sieci

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).

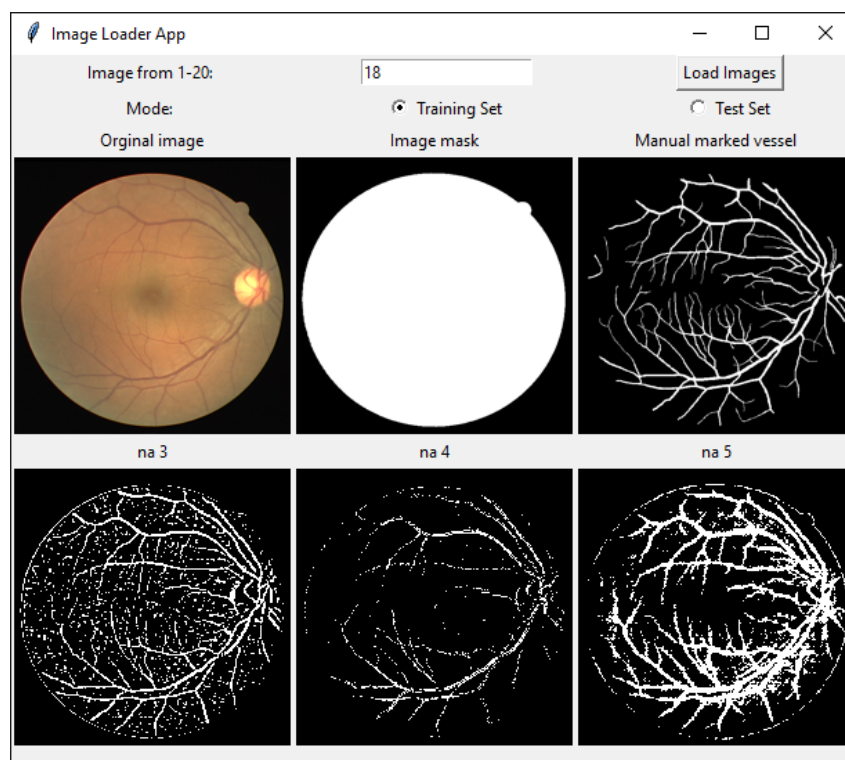
Na poniższych obrazach przedstawiono kolejno zdjęcie oka, maskę, maskę ekspercką w pierwszym rzędzie oraz kolejno wyniki działania modułów przetwarzania obrazu, uczenia maszynowego oraz sieci neuronowej w drugim (podpisano kolejno *na3*, *na4*, *na5* zgodnie z wymaganiami projektowymi).



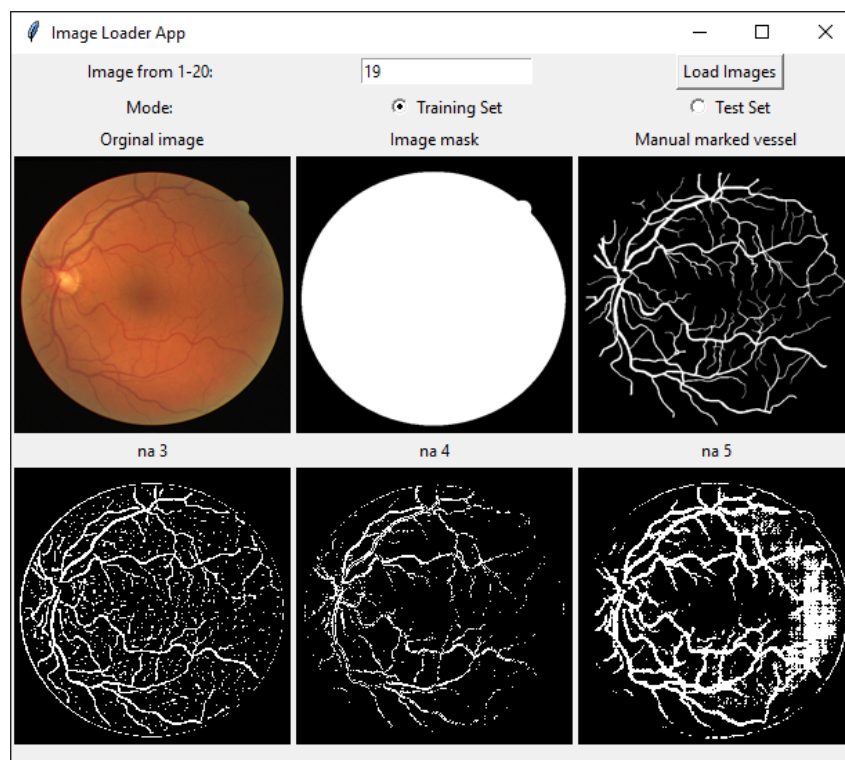
Obraz 1



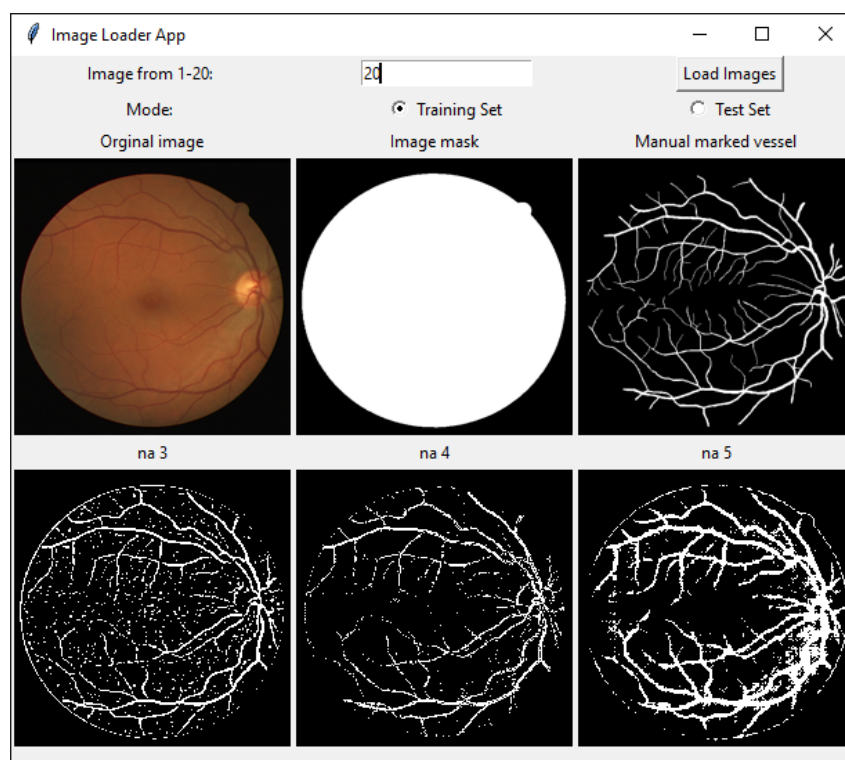
Obraz 2



Obraz 3



*Obraz 4*



*Obraz 5*

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 osiąganych przez metody przetwarzania obrazów oraz uczenia maszynowego.

- **Obraz 1**

	Trafność ( <i>accuracy</i> )	Czułość ( <i>sensitivity</i> )	Swoistość ( <i>specificity</i> )	Średnia
Przetwarzanie obrazu	0,92	0,61	0,96	0,785
Algorytm KNN	0,94	0,78	0,96	0,87
Sieć neuronowa	0,89	0,51	0,97	0,74

*Tabela 1*

Dla pierwszego obrazu wszystkie metody uzyskały bardzo dobre wyniki. Najlepszym okazał się algorytm KNN, którego trafność wynosiła 94%, osiągnął on również najwyższą czułość- najlepiej rozpoznawał naczynka (klasa pozytywna). Najlepszą swoistość osiągnęła sieć neuronowa- było to niewątpliwie związane z bardzo niskim szumem na obrazie (w pozostałych metodach pojawiało się dużo punktów pozytywnych oddzielonych od pozostałych, co zmniejszało swoistość). Średnio uczenie maszynowe metodą najbliższych sąsiadów osiągnęło o około 13% wyższe wyniki niż sieć neuronowa. Warto również zwrócić szczególną uwagę, że sieć neuronowa zaznaczała naczynka bardzo grubymi liniami- klasyfikowała więcej punktów jako klasę pozytywną, co znacznie zmniejszyło czułość. Metoda KNN zaznaczała naczynka w sposób bardziej *ostrożny*- była zdecydowanie lepsza w zaznaczaniu klasy pozytywnej i niewiele gorsza w zaznaczaniu klasy negatywnej.

- **Obraz 2**

	Trafność ( <i>accuracy</i> )	Czułość ( <i>sensitivity</i> )	Swoistość ( <i>specificity</i> )	Średnia
Przetwarzanie obrazu	0,92	0,52	0,97	0,745
Algorytm KNN	0,94	0,65	0,97	0,81
Sieć neuronowa	0,88	0,40	0,98	0,69

*Tabela 2*

W przypadku drugiego obrazu ponownie najwyższe wyniki osiągnęła metoda KNN, chociaż pozostałe dwie metody nie odbiegały znacząco w ogólnej klasyfikacji (trafności). Ponownie wszystkie metody z bardzo wysoką dokładnością zaznaczyły punkty czarne (brak naczynek, klasa negatywna). Różniła się natomiast miara czułości- względem poprzedniego obrazka, wszystkie metody uzyskały około 10% niższe wyniki. Względnie słabe klasyfikowanie naczynek (klasa pozytywna) mogło być spowodowane dużą ilością bardzo cienkich naczynek. Podobnie jak w poprzednim przykładzie, sieć neuronowa klasyfikowała nieznacznie lepiej brak naczynek,

natomiast znacznie gorzej ich obecność. Zaznaczone naczynka były zdecydowanie grubsze w przypadku sieci, powstał również pewnego rodzaju szum w okolicy największego zagęszczenia naczynek. Podobnego artefaktu nie można zaobserwować w przypadku metody KNN. Jednak przy tej metodzie (KNN) jest większy *ogólny* szum- jest to spowodowane wyznaczaniem obecności naczynek na podstawie sąsiadów. Największy szum jest widoczny w przypadku zwykłego przetwarzania obrazu.

### • Obraz 3

	Trafność ( <i>accuracy</i> )	Czułość ( <i>sensitivity</i> )	Swoistość ( <i>specificity</i> )	Średnia
Przetwarzanie obrazu	0,93	0,55	0,97	0,76
Algorytm KNN	0,94	0,81	0,94	0,875
Sieć neuronowa	0,89	0,43	0,98	0,705

Tabela 3

Trzeci obraz poddany analizie znacząco różnił się od pozostałych obrazów kolorem tła oka. Mogło to być spowodowane na przykład indywidualną chorobą oka, albo błędem przy skanie. W tym przypadku zdecydowanie najlepszą dokładność uzyskał algorytm KNN, który zaznaczył najmniej punktów białych (naczynek)- znacząco mniej od pozostałych. W przypadku klasycznego przetwarzania obrazu powstał typowy szum, natomiast same naczynka były zaznaczone prawidłowo. Algorytm sieci neuronowej zaznaczył, podobnie jak w poprzednich przykładach, bardzo grubo naczynka, co prowadziło do znacznego spadku czułości, a tym samym średniej, która była o ponad 0,1 niższa niż średnia przetwarzania obrazu. Metoda KNN przyporządkowując bardzo mało punktów do klasy pozytywnej uzyskała o około 17% lepsze rezultaty (po uśrednieniu czułości i swoistości). Jest to prawdopodobnie spowodowane niższym kontrastem pomiędzy naczynkami a tłem oka.

### • Obraz 4

	Trafność ( <i>accuracy</i> )	Czułość ( <i>sensitivity</i> )	Swoistość ( <i>specificity</i> )	Średnia
Przetwarzanie obrazu	0,93	0,57	0,96	0,765
Algorytm KNN	0,94	0,73	0,98	0,855
Sieć neuronowa	0,87	0,39	0,98	0,685

Tabela 4

W przypadku czwartego przykładu metody przetwarzania obrazu oraz wyznaczania klasy metodą najbliższych sąsiadów poradziły się porównywalnie do poprzednich obrazów, natomiast widać znaczące pogorszenie klasyfikacji dokonanej przez sieć neuronową. Powstał znaczący artefakt po prawej stronie- znacząca ilość punktów została błędnie sklasyfikowana jako klasa pozytywna, co prowadziło do znacznego pogorszenia czułości (prawie 20% względem przetwarzania obrazu oraz prawie 35% względem metody KNN). Mógł on być spowodowany zagęszczeniem drobnych naczynek, w szczególności nietypowym ich *skrzyżowaniem* po prawej

stronie oka. Ponownie powstał spory szum w metodzie przetwarzania obrazów. Sieć neuronowa zaznaczyła naczynka bardzo grubymi liniami.

- **Obraz 5**

	Trafność ( <i>accuracy</i> )	Czułość ( <i>sensitivity</i> )	Swoistość ( <i>specificity</i> )	Średnia
Przetwarzanie obrazu	0,94	0,57	0,98	0,775
Algorytm KNN	0,95	0,72	0,98	0,85
Sieć neuronowa	0,90	0,42	0,99	0,705

*Tabela 5*

Ostatni z analizowanych przypadków charakteryzował się nieco ciemniejszym tłem oka oraz dużą ilością bardzo małych, prawie niewidocznych naczynek. Metoda przetwarzania obrazu uzyskała bardzo dobre wyniki, nieznacznie lepsze niż w pozostałych przypadkach. Prawdopodobnie część wygenerowanego szumu *trafiła* poprawnie w naczynka. Algorytm KNN wygenerował względnie czysty rysunek (bez szumu), jednak nie zaznaczył wszystkich drobnych naczynek. Bardzo dobrze poradził sobie przy największym zagęszczeniu naczynek (po prawej stronie oka). Sieć neuronowa uzyskała względnie dobre wyniki, czułość była dużo mniejsza niż w przypadku uczenia maszynowego metodą KNN, a swoistość nieznacznie wyższa (było to spowodowane grubymi liniami- wiele punktów dookoła naczynek zostało uznane błędnie jako klasa pozytywna). Sieć poradziła sobie również dużo gorzej w największym zagęszczeniu, gdzie został wygenerowany spory szum.

## 6. Podsumowanie

W analizowanych przykładach wszystkie metody rozpoznawania naczynek oka poradziły sobie dosyć dobrze z zadaniem klasyfikacji. Dla metody przetwarzania obrazu charakterystyczny był duży szum oraz względnie dobre zaznaczanie drobnych naczynek. Metoda KNN generowała najlepsze rezultaty, które nieraz znacząco przewyższały pozostałe metody. Charakterystyczne było bardzo dokładne zaznaczanie dużych naczynek oraz względnie niewielki szum. Wyszukiwanie naczynek przy użyciu sieci neuronowej przynosiło dobre rezultaty, zważywszy niewielki zbiór treningowy użyty do uczenia sieci. Często generowane były bardzo grube naczynka, prawdopodobnie było to spowodowane architekturą sieci, która wymagała częstej zmiany wymiarów obrazka. Powstawał również spory szum w miejscach, gdzie było największe zagęszczenie, albo gdzie pojawiały się nietypowe połączenia.

W rzeczywistym zastosowaniu najlepszym podejściem byłoby wykorzystanie wszystkich metod oraz ich połączenie nadzorowane manualnie. Takie podejście pozwoliłoby uzyskać najwyższe rezultaty, wykluczając wszelkie artefakty.