

Raport - wykrywanie naczyń dna siatkówki oka

1 Skład grupy

- Zuzanna Piniarska 136782
- Mateusz Kałamoniak 136730

2 Zastosowany język programowania i biblioteki

Język programowania: **Python**

Dodatkowo zastosowane biblioteki

- tensorflow
- numpy
- PIL
- matplotlib
- datetime
- IPython

3 Opis zastosowanych metod

W implementacji posłużyliśmy się w głównej mierze biblioteką Tensorflow. Posłużyła ona do wstępnego przetwarzania obrazu jak i tworzenia sieci neuronowej. Zastosowaliśmy techniki augmentacji takie jak modyfikacja saturacji, kontrastu czy odcienia. Wszystkim z tych funkcji podaliśmy losowe wartości w określonych przedziałach. Dodatkowo obracaliśmy obraz. Z użyciem tak przetworzonych obrazów wytrenowaliśmy sieć neuronową. Zaimplementowaliśmy sieć będącą modyfikacją U-Net i MobileNetV2. Wykorzystaliśmy architekturę z U-Net, natomiast encoder został zaimplementowany

na wzór MobileNetV2. Do optymalizacji zastosowaliśmy algorytm Adam z współczynnikiem uczenia 0.001. Funkcja kosztu składa się z sumy indeksu Jaccarda z wagą 0.3 oraz binary cross entropy z wagą 0.7. Przy tworzeniu sieci sugerowaliśmy się przede wszystkim artykułem M2U-Net: Effective and Efficient Retinal Vessel Segmentation for Real-World Applications (<https://arxiv.org/pdf/1811.07738.pdf>).

Uzupełnione informacje

Nasza sieć uczyła się przez około 3 godziny, uczyła się na 200 epokach. Ze względu na ilość obrazów wejściowych podzieliliśmy nasz zbiór na 15 obrazów treningowych i 30 obrazów walidujących. Obrazy pochodziły ze zbioru HRF.

Wykorzystaliśmy augmentację 15 obrazów składającą się ze zmiany saturacji, kontrastu i odcienia oraz orientacji.

Rozmiary zdjęć są zmienione na 544x544. Pomijając powyższe zmiany resztę algorytmu staraliśmy się zaimplementować sugerując się pracą naukową.

Podstawowe przetwarzanie:

W podstawowym przetwarzaniu zastosowaliśmy normalizację histogramu za pomocą algorytmu CLAHE (Contrast Limited Adaptive Histogram Equalization), filtr Gaussa, adaptive threshold wyliczony metodą Gaussa oraz dylatacja. Wyniki można uznać za zadowalające jednak sieć neuronowa dużo lepiej poradziła sobie z zadaniem.

4 Wyniki parametrów przetwarzania

4.1 Parametry przetwarzania za pomocą sieci neuronowej

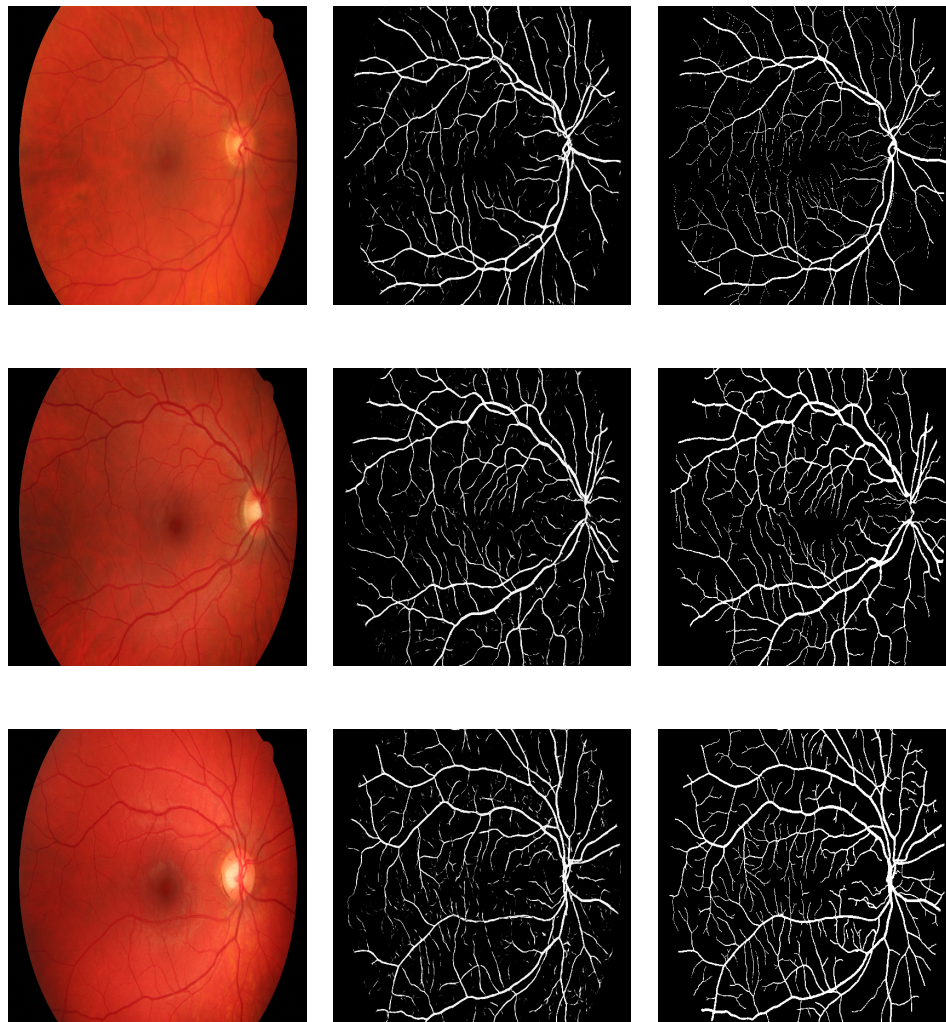
| Name | Accuracy | Precision | Sensitivity | Specificity | F1 score |
|-------|----------|-----------|-------------|-------------|----------|
| 01_dr | 0,9312 | 0,3599 | 0,4523 | 0,9569 | 0,4008 |
| 01_g | 0,9340 | 0,5086 | 0,4260 | 0,9705 | 0,4637 |
| 01_h | 0,8980 | 0,4992 | 0,4211 | 0,9521 | 0,4568 |
| 02_dr | 0,9226 | 0,3996 | 0,4312 | 0,9560 | 0,4148 |
| 02_g | 0,9317 | 0,5495 | 0,4587 | 0,9697 | 0,5000 |
| 03_h | 0,9046 | 0,5750 | 0,3780 | 0,9669 | 0,4562 |
| 04_dr | 0,9125 | 0,2919 | 0,3899 | 0,9436 | 0,3339 |
| 04_g | 0,9396 | 0,5417 | 0,4041 | 0,9765 | 0,4629 |
| 04_h | 0,9143 | 0,5615 | 0,4146 | 0,9663 | 0,4770 |
| 05_dr | 0,9367 | 0,4917 | 0,4862 | 0,9666 | 0,4889 |
| 05_g | 0,9433 | 0,5842 | 0,4545 | 0,9774 | 0,5113 |
| 07_g | 0,9173 | 0,3861 | 0,4559 | 0,9494 | 0,4181 |
| 07_h | 0,9256 | 0,6257 | 0,4149 | 0,9757 | 0,4990 |
| 08_g | 0,9106 | 0,3763 | 0,4869 | 0,9414 | 0,4245 |
| 09_dr | 0,9251 | 0,4309 | 0,3310 | 0,9683 | 0,3744 |
| 09_h | 0,9269 | 0,5419 | 0,3800 | 0,9729 | 0,4467 |
| 10_dr | 0,9241 | 0,5930 | 0,4228 | 0,9722 | 0,4936 |
| 10_g | 0,9276 | 0,4466 | 0,4435 | 0,9615 | 0,4450 |
| 11_g | 0,9164 | 0,4492 | 0,4335 | 0,9562 | 0,4412 |
| 11_h | 0,9259 | 0,6491 | 0,4516 | 0,9748 | 0,5326 |
| 12_dr | 0,9240 | 0,4351 | 0,3970 | 0,9624 | 0,4152 |
| 12_h | 0,9182 | 0,6575 | 0,4586 | 0,9720 | 0,5403 |
| 13_dr | 0,9222 | 0,4230 | 0,3800 | 0,9620 | 0,4004 |
| 13_g | 0,9212 | 0,4183 | 0,3987 | 0,9595 | 0,4083 |
| 13_h | 0,9229 | 0,5689 | 0,4593 | 0,9669 | 0,5083 |
| 14_dr | 0,9057 | 0,3926 | 0,4309 | 0,9450 | 0,4109 |
| 14_h | 0,9255 | 0,5916 | 0,4647 | 0,9694 | 0,5206 |
| 15_dr | 0,9109 | 0,3545 | 0,4541 | 0,9426 | 0,3981 |
| 15_g | 0,9303 | 0,5351 | 0,4000 | 0,9724 | 0,4578 |
| 15_h | 0,9348 | 0,6129 | 0,4499 | 0,9759 | 0,5189 |

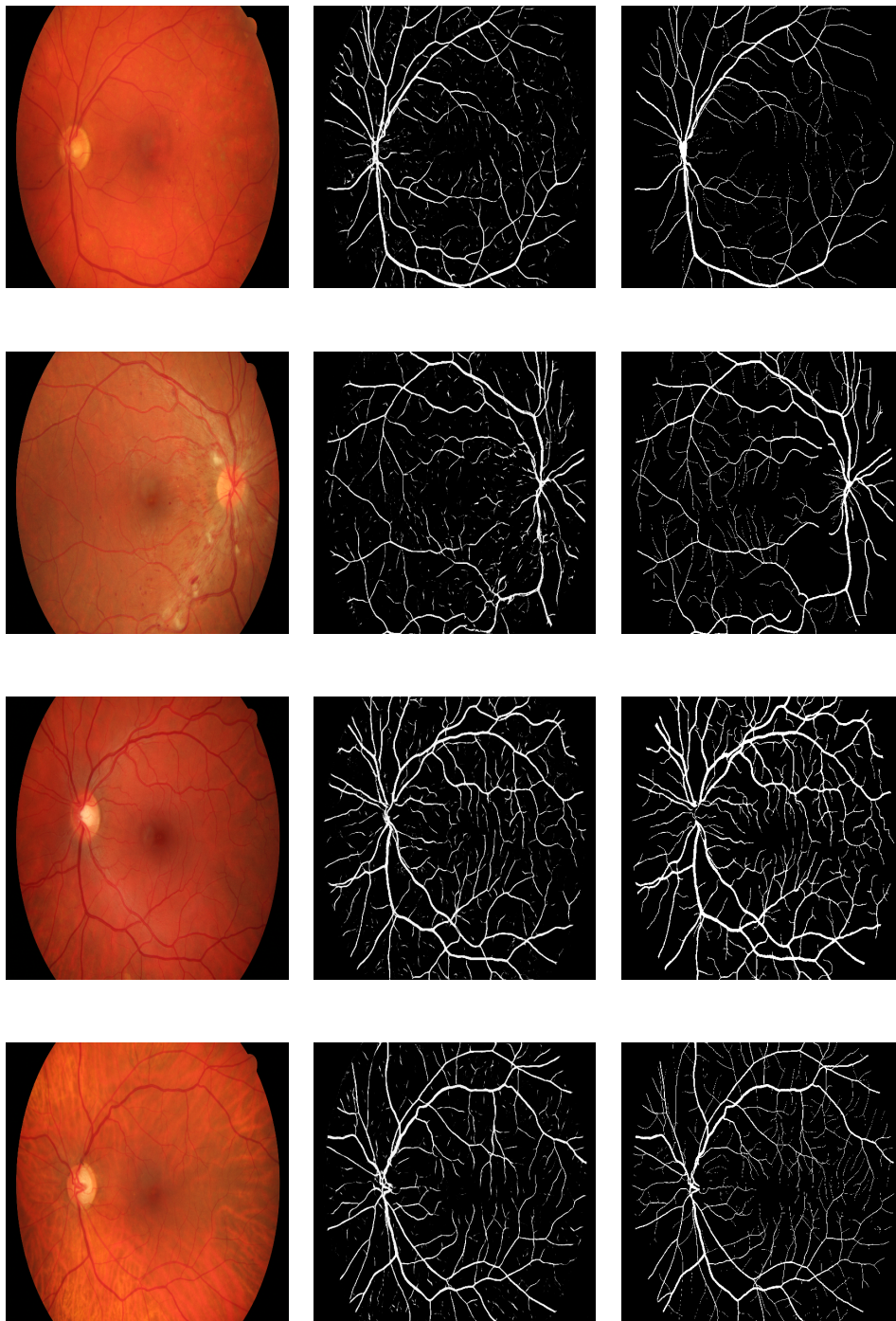
4.2 Parametry przetwarzania przy pomocy przetwarzania obrazu

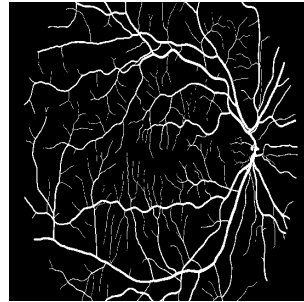
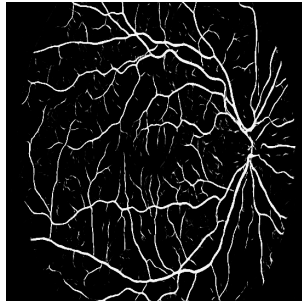
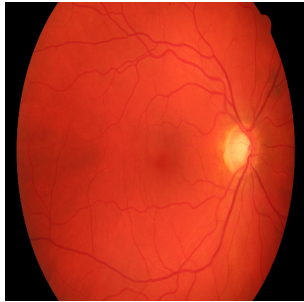
| Name | Accuracy | Precision | Sensitivity | Specificity | F1 score |
|-----------|----------|-----------|-------------|-------------|----------|
| 01_h.jpg | 0,9514 | 0,8387 | 0,6479 | 0,9858 | 0,7311 |
| 12_dr.jpg | 0,9583 | 0,7706 | 0,5526 | 0,9880 | 0,6436 |
| 14_h.jpg | 0,9576 | 0,8039 | 0,6797 | 0,9842 | 0,7366 |
| 13_h.jpg | 0,9577 | 0,8107 | 0,6707 | 0,9851 | 0,7341 |
| 10_g.jpg | 0,9624 | 0,7173 | 0,7056 | 0,9804 | 0,7114 |
| 01_dr.jpg | 0,9574 | 0,5605 | 0,7552 | 0,9683 | 0,6435 |
| 03_h.jpg | 0,9450 | 0,8120 | 0,6247 | 0,9829 | 0,7061 |
| 15_dr.jpg | 0,9576 | 0,6767 | 0,6738 | 0,9775 | 0,6753 |
| 11_h.jpg | 0,9587 | 0,8296 | 0,7021 | 0,9851 | 0,7605 |
| 04_g.jpg | 0,9630 | 0,7082 | 0,7285 | 0,9792 | 0,7182 |
| 15_h.jpg | 0,9631 | 0,8529 | 0,6394 | 0,9906 | 0,7309 |
| 09_h.jpg | 0,9599 | 0,7712 | 0,6876 | 0,9828 | 0,7270 |
| 09_dr.jpg | 0,9558 | 0,7361 | 0,5430 | 0,9858 | 0,6250 |
| 07_h.jpg | 0,9600 | 0,7961 | 0,7407 | 0,9814 | 0,7674 |
| 04_dr.jpg | 0,9666 | 0,7253 | 0,6588 | 0,9850 | 0,6905 |
| 13_g.jpg | 0,9561 | 0,6746 | 0,6856 | 0,9759 | 0,6801 |
| 05_dr.jpg | 0,9576 | 0,6272 | 0,7891 | 0,9688 | 0,6989 |
| 10_dr.jpg | 0,9488 | 0,7036 | 0,7102 | 0,9715 | 0,7069 |
| 05_g.jpg | 0,9653 | 0,7313 | 0,7440 | 0,9808 | 0,7376 |
| 02_dr.jpg | 0,9571 | 0,6476 | 0,7219 | 0,9731 | 0,6827 |
| 08_g.jpg | 0,9603 | 0,6889 | 0,7587 | 0,9750 | 0,7221 |
| 01_g.jpg | 0,9558 | 0,6471 | 0,7507 | 0,9706 | 0,6950 |
| 13_dr.jpg | 0,9618 | 0,8079 | 0,5762 | 0,9900 | 0,6727 |
| 02_g.jpg | 0,9566 | 0,6945 | 0,7396 | 0,9739 | 0,7164 |
| 04_h.jpg | 0,9539 | 0,7793 | 0,7109 | 0,9791 | 0,7435 |
| 12_h.jpg | 0,9558 | 0,8413 | 0,7119 | 0,9843 | 0,7712 |
| 15_g.jpg | 0,9552 | 0,6791 | 0,7296 | 0,9729 | 0,7035 |
| 07_g.jpg | 0,9632 | 0,7111 | 0,7339 | 0,9792 | 0,7223 |
| 14_dr.jpg | 0,9547 | 0,7522 | 0,6057 | 0,9835 | 0,6710 |
| 11_g.jpg | 0,9578 | 0,7207 | 0,7336 | 0,9764 | 0,7271 |

5 Wizualizacja i analiza wyników działania programu

W poniżej przedstawionych wynikach po lewej stronie znajduje się oryginalne zdjęcie, po środku predykcja, a po prawej maska.

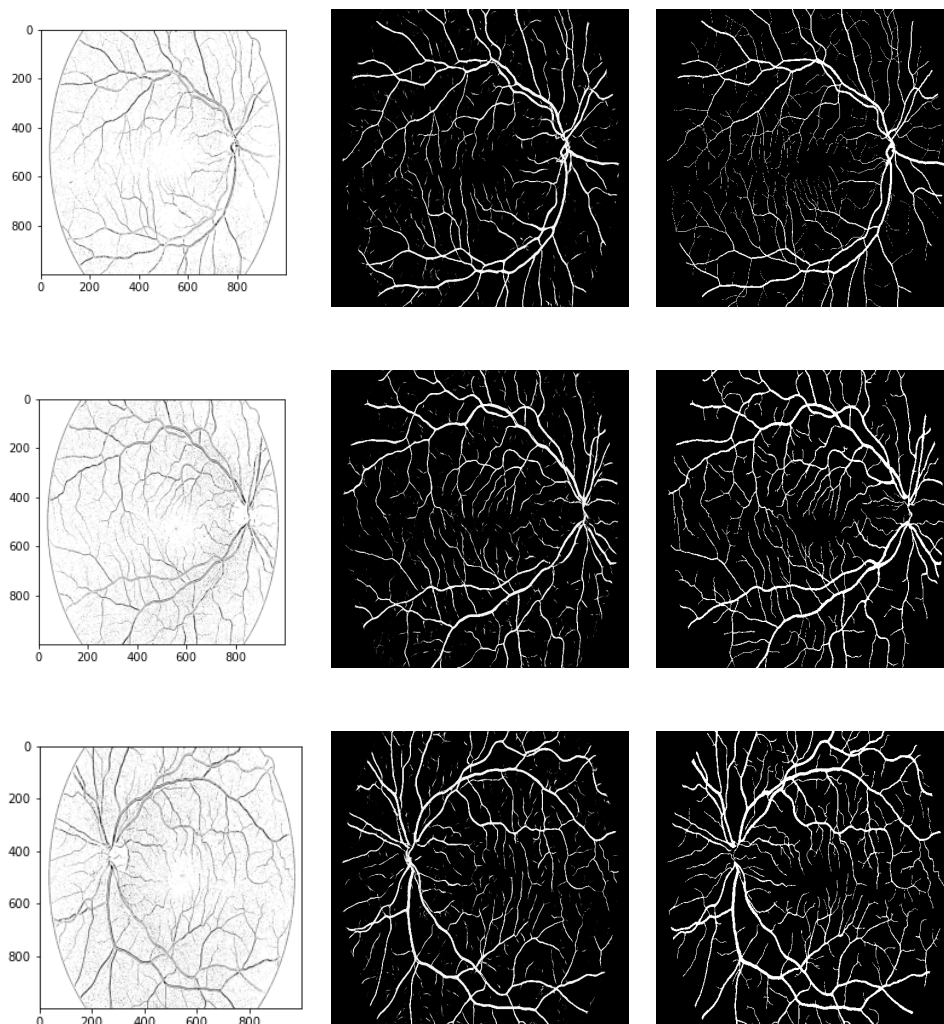






5.1 Porównanie działania sieci neuronowej i zwykłego przetwarzania obrazu

Po lewej stronie znajduje się efekt prostego przetwarzania obrazu, po środku wynik działania sieci neuronowej, a po prawej - maska.



6 Podsumowanie

Wyniki dla prostego przetwarzania obrazów i sieci neuronowej należy interpretować mając na uwadze wysokie niezrównoważenie klas wynoszące 7,6%.

Zdecydowanie więcej pikseli należy do klasy TRUE oznaczającej naczynia krwionośne, niż do FALSE oznaczającej tło. Accuracy dla prostego przetwarzania na poziomie 92,5% przy takim poziomie niezrównoważnia klas oznacza, że połowa pikseli naczyń krwionośnych została błędnie rozpoznana. Potwierdza to współczynnik Precision o średnim wyniku 49,5%, który mówi o tym ile procent pikseli zaklasyfikowanych do klasy TRUE tak naprawdę do niej należy. Dla sieci neuronowej Precision na poziomie 73,3% jest bardzo dobrym wynikiem pozwalającym uznać, że zdecydowana większość naczyń krwionośnych została poprawnie rozpoznana. Przez niebalansownię klas, wysoka średnia swoistość dla obu podejść nie jest zatem niczym niespodziewanym. Dużo ważniejsza jest czułość i F1-score. Czułość na poziomie 42,7% świadczy o bardzo niskim procencie trafień z poprawnym rozpoznaniem naczynia krwionośnego jako naczynie krwionośne. Ponownie dla sieci neuronowej statystyki są zdecydowanie lepsze, ponad 2/3 pikseli z klasy TRUE zostało poprawnie rozpoznanych. F1-score jest średnią harmoniczną czułości i precyzji, dlatego ze względu na podobne wyniki obu miar, również jest do nich podobny.

| | Accuracy | Precision | Sensitivity | Specificity | F1 score |
|-------------------|----------|-----------|-------------|-------------|----------|
| Processing | 0,9228 | 0,4950 | 0,4277 | 0,9634 | 0,4540 |
| M2U-Net | 0,9578 | 0,7372 | 0,6904 | 0,9799 | 0,7084 |