



POLITECHNIKA POZNAŃSKA

Wydział Informatyki i Telekomunikacji

Wykrywanie naczyń dna siatkówki oka

Szymon Jakubiec

nr indeksu 126997

Damian Cybulski

nr indeksu 120573

Prowadzący:

dr hab. inż. Szymon Wilk

Język programowania

Zastosowanym przez nas językiem programowania był **Python**, który oferował odpowiednie biblioteki, ułatwiające przetwarzanie obrazu. Dodatkowym atutem była możliwość wykorzystania **Jupyter Notebooka**. Wykorzystano następujące biblioteki:

- Skimage
- Matplotlib
- Sklearn
- Imblearn

Opis zastosowanych metod

Przetwarzanie obrazów

Przetwarzanie obrazów wejściowych zostało wykonane wieloetapowo w następujących krokach:

1. Wczytanie obrazu wejściowego i eksperckiego

```
def load_base_image(self, base_image):  
    self.base_image = base_image  
  
def load_expert_image(self, expert_image):  
    self.expert_image = expert_image
```

2. Przefiltrowanie obrazu i wyodrębnienie barwy zielonej

```
def extract_green(self):  
    self.green_image = self.base_image.copy()  
    self.green_image[:, :, 0] = 0  
    self.green_image[:, :, 2] = 0
```

3. Konwersja obrazu do skali szarości

```
def convert_to_gray(self):  
    self.gray_image = skimage.color.rgb2gray(self.green_image.copy())
```

4. Normalizacja histogramu kolorów

```
def normalize(self):  
    self.normalized_image = exposure.equalize_hist(self.gray_image.copy())  
    self.normalized_image = rank.equalize(skimage.util.img_as_ubyte(self.normalized_image), selem=disk(9))
```

5. Wykrywanie naczyń krwionośnych przy wykorzystaniu filtra Frangi'ego

```
def frangi_filter(self):  
    K = ones([14, 14])  
    K = K / sum(K)  
    self.frangied_image = convolve(fragi(self.rescaled_image), K)
```

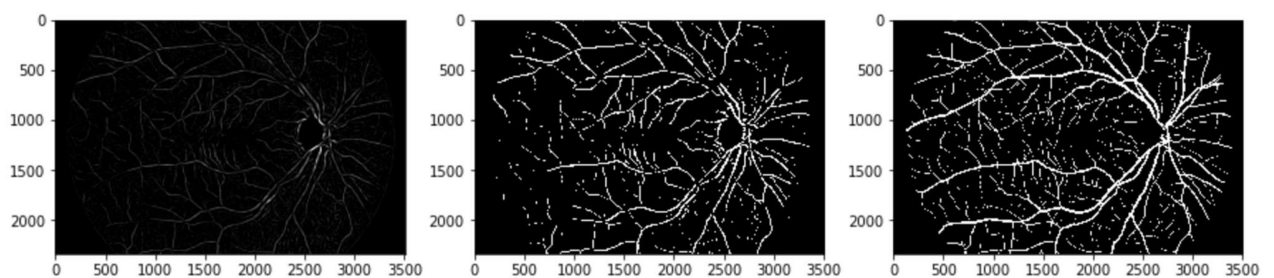
6. Odcięcie tła

```
def background_cut_off(self):  
    self.binary_image = self.frangied_image > np.mean(self.frangied_image) + 1.5 * np.std(self.frangied_image)
```

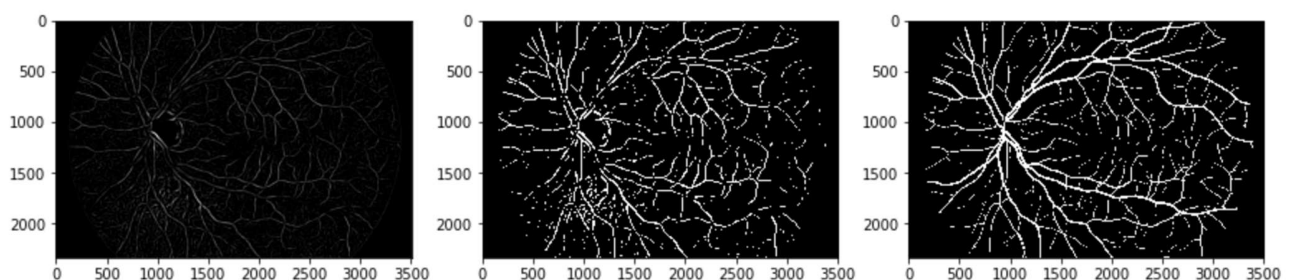
Wizualizacja wyników

Zaimplementowany algorytm wykrywania naczyń krwionośnych zastosowano na 5 wejściowych obrazach. Poniżej przedstawiono wizualizację wyników.

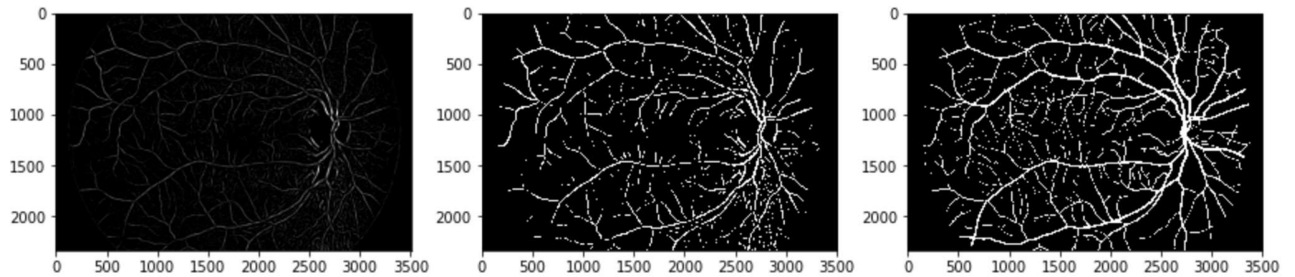
Obraz 1 - od lewej - obraz po zastosowaniu filtra Frangi'ego, obraz z zastosowanym odcięciem tła, obraz ekspercki



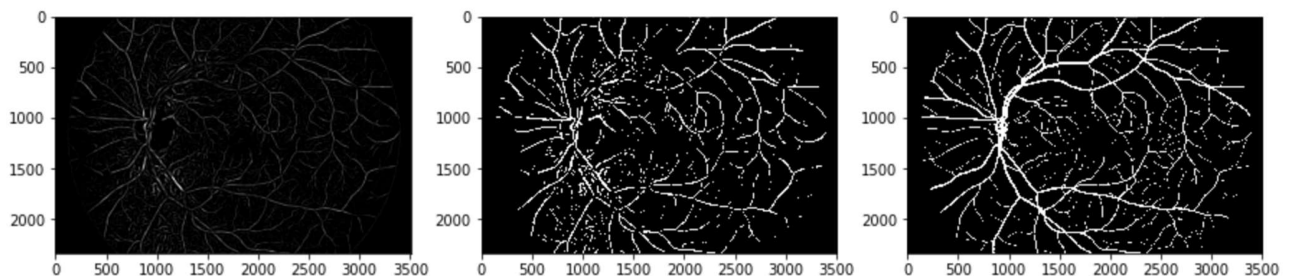
Obraz 2 - od lewej - obraz po zastosowaniu filtra Frangi'ego, obraz z zastosowanym odcięciem tła, obraz ekspercki



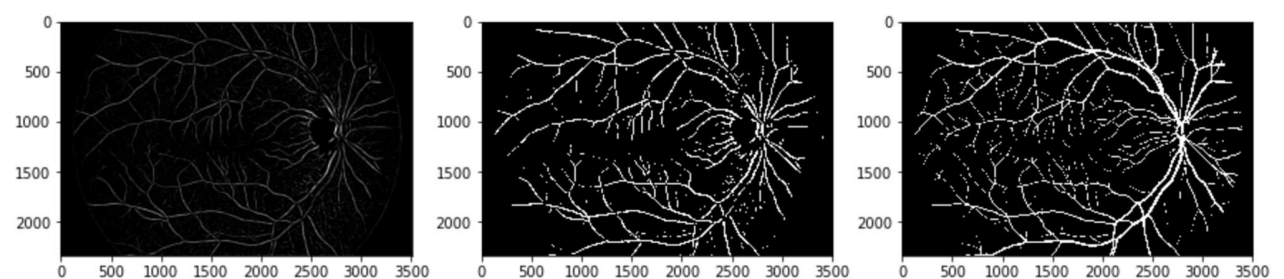
Obraz 3 - od lewej - obraz po zastosowaniu filtra Frangi'ego, obraz z zastosowanym odcięciem tła, obraz ekspercki



Obraz 4 - od lewej - obraz po zastosowaniu filtra Frangi'ego, obraz z zastosowanym odcięciem tła, obraz ekspercki



Obraz 5 - od lewej - obraz po zastosowaniu filtra Frangi'ego, obraz z zastosowanym odcięciem tła, obraz ekspercki



Ocena statystyczna wyników

Oceny statystycznej wygenerowanych obrazów dokonano przy użyciu dostępnych bibliotek `sklearn.metrics` oraz `imblearn.metrics`. Badano takie parametry jak dokładność, czułość, swoistość pomiaru. Dodatkowo wykorzystano miary statystyczne dla danych niezrównoważonych takie jak miara `f1`, `precision` oraz `recall`. Wyniki oraz funkcję obliczającą efektywność algorytmu przedstawiono poniżej.

```

def calculate_effectiveness(self):
    img = self.binary_image.astype(np.uint8)
    img = np.array(img).reshape(-1)
    expert_img = self.expert_image
    expert_img = np.divide(expert_img, 255).astype(np.uint8)
    expert_img = np.array(expert_img).reshape(-1)
    print("Classification Report Imbalanced")
    print(classification_report_imbalanced(img, expert_img))

    conf_matrix = confusion_matrix(expert_img, img)
    true_positive = conf_matrix[0][0]
    false_positive = conf_matrix[0][1]
    false_negative = conf_matrix[1][0]
    true_negative = conf_matrix[1][1]
    print("TruePositive: {}, FalsePositive: {}, FalseNegative: {}, TrueNegative:{}".format(true_positive, false_positive, false_negative, true_negative))

    accuracy = (true_positive + true_negative) / (true_positive + true_negative + false_positive + false_negative)
    sensitivity = true_positive / (true_positive + false_negative)
    specificity = true_negative / (true_negative + false_positive) if (true_negative + false_positive) != 0 else 0
    print("Accuracy: {}, Sensitivity: {}, Specificity: {}".format(accuracy, sensitivity, specificity))

    f1 = f1_score(img, expert_img)
    print("F1_score: {}".format(f1))

```

Obraz 1 - metryki

Classification Report Imbalanced							
	pre	rec	spe	f1	geo	iba	sup
0	0.98	0.95	0.75	0.96	0.84	0.73	7598638
1	0.53	0.75	0.95	0.62	0.84	0.70	586706
avg / total	0.95	0.93	0.76	0.94	0.84	0.72	8185344

TruePositive: 7204656, FalsePositive: 146800, FalseNegative: 393982, TrueNegative:439906
 Accuracy: 0.933932892740977, Sensitivity: 0.9481509712661664, Specificity: 0.7497895027492475
 F1_score: 0.6193268449676684

Obraz 2 - metryki

Classification Report Imbalanced							
	pre	rec	spe	f1	geo	iba	sup
0	0.98	0.95	0.71	0.96	0.83	0.70	7552357
1	0.56	0.71	0.95	0.63	0.83	0.66	632987
avg / total	0.94	0.93	0.73	0.94	0.83	0.69	8185344

TruePositive: 7196263, FalsePositive: 180643, FalseNegative: 356094, TrueNegative:452344
 Accuracy: 0.9344270686729843, Sensitivity: 0.9528499513463148, Specificity: 0.7146181517155961
 F1_score: 0.6276344589555475

Obraz 3 - metryki

Classification Report Imbalanced							
	pre	rec	spe	f1	geo	iba	sup
0	0.98	0.95	0.76	0.96	0.85	0.73	7582165
1	0.53	0.76	0.95	0.62	0.85	0.70	603179
avg / total	0.95	0.93	0.77	0.94	0.85	0.73	8185344

TruePositive: 7173430, FalsePositive: 147743, FalseNegative: 408735, TrueNegative:455436
Accuracy: 0.9320153190873836, Sensitivity: 0.946092573822912, Specificity: 0.7550594433824785
F1_score: 0.6207598732408764

Obraz 4 - metryki

Classification Report Imbalanced							
	pre	rec	spe	f1	geo	iba	sup
0	0.98	0.96	0.71	0.97	0.82	0.69	7562510
1	0.57	0.71	0.96	0.63	0.82	0.66	622834
avg / total	0.94	0.94	0.73	0.94	0.82	0.69	8185344

TruePositive: 7231864, FalsePositive: 181879, FalseNegative: 330646, TrueNegative:440955
Accuracy: 0.9373850384296616, Sensitivity: 0.9562782726898873, Specificity: 0.7079815809669993
F1_score: 0.6324497018505704

Obraz 5 - metryki

Classification Report Imbalanced							
	pre	rec	spe	f1	geo	iba	sup
0	0.98	0.96	0.76	0.97	0.86	0.75	7561837
1	0.64	0.76	0.96	0.69	0.86	0.72	623507
avg / total	0.95	0.95	0.77	0.95	0.86	0.75	8185344

TruePositive: 7296272, FalsePositive: 150286, FalseNegative: 265565, TrueNegative:473221
Accuracy: 0.9491956599502721, Sensitivity: 0.9648808880699227, Specificity: 0.7589666194605674
F1_score: 0.6947418800507673