

Tęczówka oka

Michał Matuszyk, Paweł Pozorski

Kwiecień 2025

Abstract

W niniejszym projekcie zrealizowano pełną ścieżkę przetwarzania obrazu tęczówki oka w celu wygenerowania charakterystycznego „kodu tęczówki” zgodnie z podejściem zaproponowanym przez Johna Daugmana. Przeprowadzone testy wstępne potwierdziły poprawność działania algorytmu oraz wykazały obiecujące rezultaty w kontekście zastosowań biometrycznych.

Contents

1	Wprowadzenie	3
2	Przygotowanie obrazu tęczówki	4
2.1	Zbiór danych	4
2.2	Detekcja granic źrenicy	4
2.3	Detekcja granic tęczówki	9
2.4	Ekstrakcja tęczówki	12
3	Algorytm Daugmana	15
4	Demo	16
5	Podsumowanie	17

1 Wprowadzenie

Demonstracja całego procesu

Najciekawsze etapy procesu są zilustrowane na Figure 1.

Pierwszym etapem jest **dokładna segmentacja tęczówki**. Proces ten musi być odporny na odbicia światła, zmienny kąt rejestracji czy różnorodne rozmiary źrenicy, dlatego wymaga starannego doboru filtrów i progów.

Po wyznaczeniu brzegów tęczówki obraz zostaje **przetransformowany ze współrzędnych biegunowych do kartezjańskich**. Rozwinięty pierścień dzieli się na **osiem poziomych pasów**, z których każdy podlega selektywnemu przycięciu, tak aby usunąć fragmenty zniekształcone lub pozbawione informacji.

Następnie pasy są **skomprimowane** do rozmiaru 1×128 px (szerokość \times wysokość), co stabilizuje rozdzielczość wejściową i upraszcza dalsze obliczenia.

Na tak przygotowanych danych uruchamiany jest **algorytm Daugmana**, który przy użyciu filtrów Gabora generuje binarny *kod tęczówki* – końcowy, zwarty i porównywalny pomiędzy osobami.

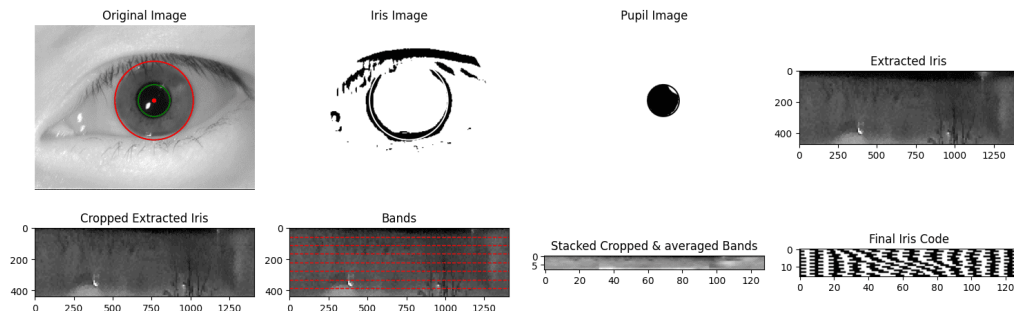


Figure 1: Skrótowy pokaz całego procesu

2 Przygotowanie obrazu tęczówki

2.1 Zbiór danych

Zbiór danych tęczówek wykorzystany w analizie pochodzi z platformy Kaggle i nosi nazwę *naureenmohammad/mmu-iris-dataset*. Zawiera on zdjęcia oczu (prawego i lewego) pochodzące od 45 osób, z których każde oko zostało sfotografowane pięciokrotnie. Niestety, zdjęcia cechują się jedynie przeciętną jakością – w wielu przypadkach znaczna część tęczówki jest przesłonięta przez powiekę. Przykład takiej sytuacji przedstawiono na Figure 2, gdzie fragment tęczówki jest zakryty zarówno od góry, jak i od dołu. Przykład 10 zdjęć pochodzących od jednej osoby jest widoczny na Figure 3. Jak widać, zdjęcia różnią się między sobą i rzadko źrenica jest na środku.

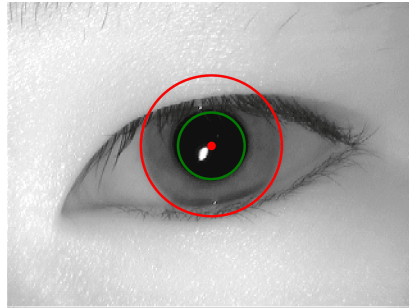


Figure 2: Znaczna część tęczówki jest zakryta przez powiekę zarówno górną jak i dolną.

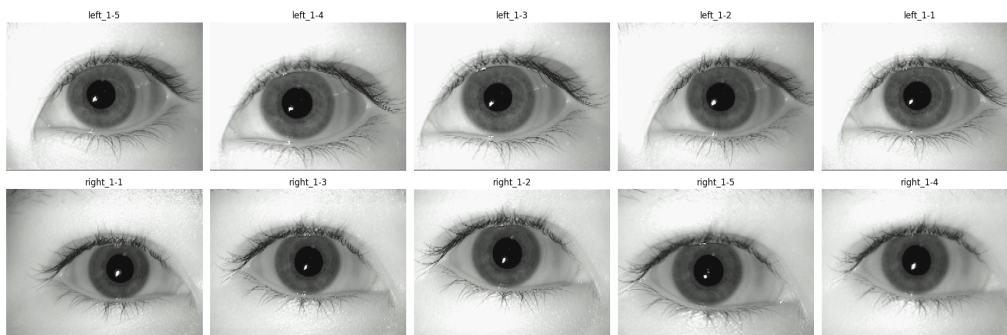


Figure 3: Przykłady tęczówek

2.2 Detekcja granic źrenicy

Detekcja granic źrenicy stanowi kluczowy etap wstępnego przetwarzania obrazu tęczówki. W badaniu zastosowano proste podejście progowe, oparte na średniej jasności obrazu. Dla każdego obrazu obliczana jest średnia wartość piksela, a następnie stosowane są różne wartości progowe będące jej ułamkami, co umożliwia segmentację potencjalnego obszaru źrenicy. Przykład działania tej metody przedstawiono na Figure 4, gdzie widoczne są wyniki progowania obrazu dla różnych wartości progowych.

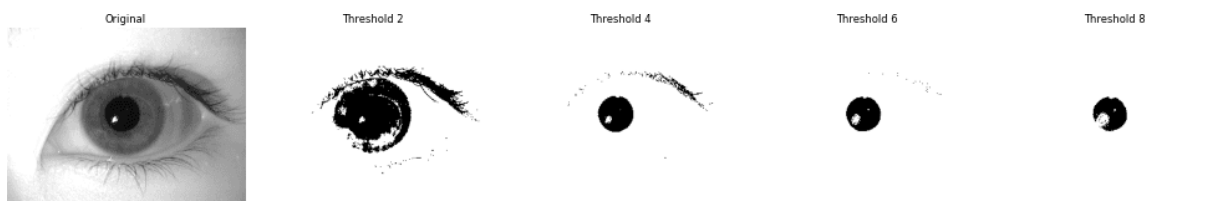


Figure 4: Przykład progowania obrazu w celu detekcji źrenicy dla różnych wartości progów.

Dla lepszego zobrazowania działania algorytmu, na ilustracji Figure 5 przedstawiono efekty progowania dla losowo wybranych obrazów. Widać, że skuteczność detekcji zależy od jakości obrazu oraz parametrów progowania.

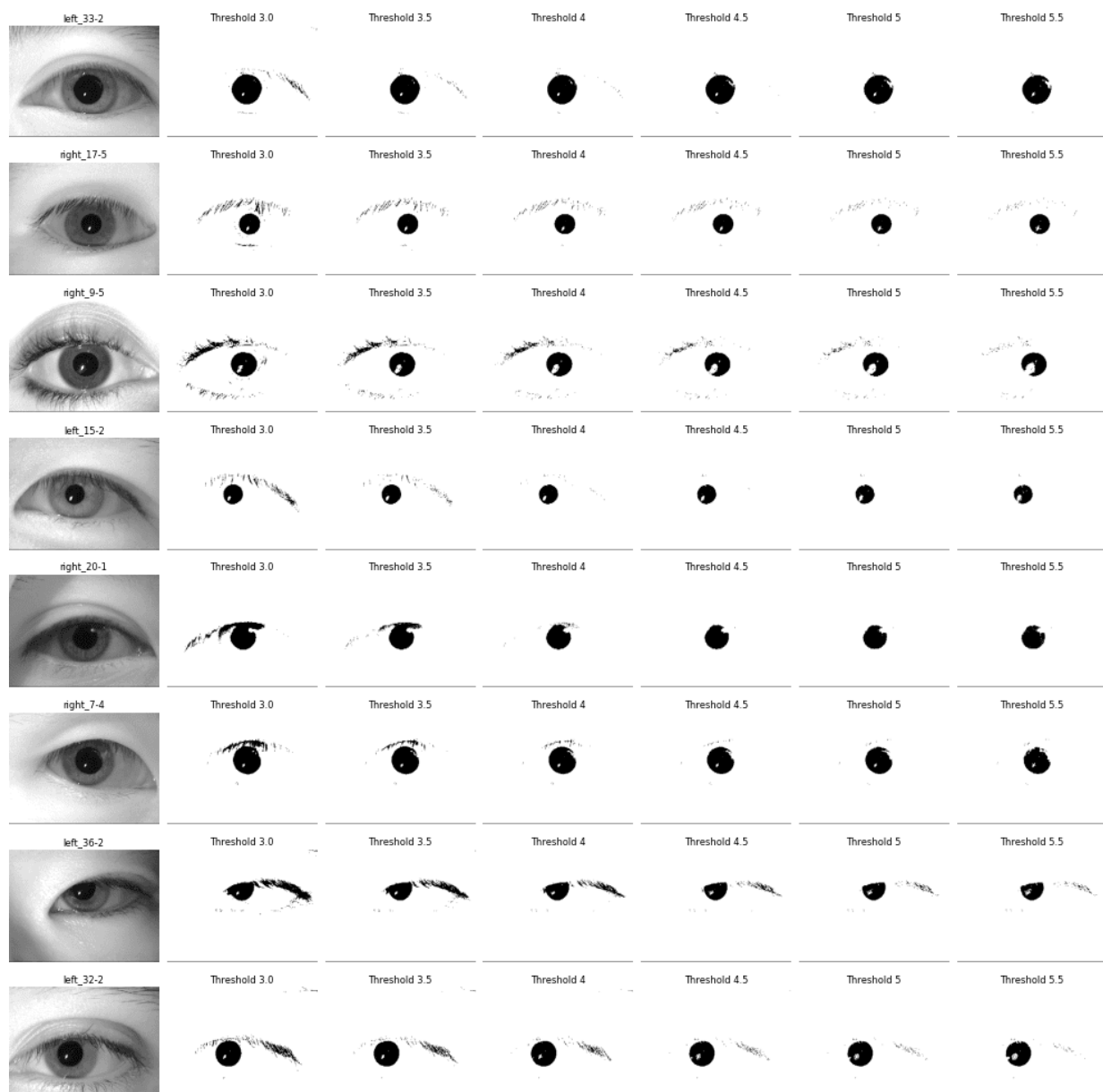


Figure 5: Progowanie źrenicy dla różnych zdjęć i progów.

W celu poprawy jakości detekcji źrenicy, zastosowano filtr wyostrzający w połączeniu z regulacją kontrastu. Wyniki tej operacji pokazano na Figure 6. Zmodyfikowane obrazy wykazują większy kontrast w obszarze źrenicy, co umożliwia dokładniejsze wyznaczenie jej granic.

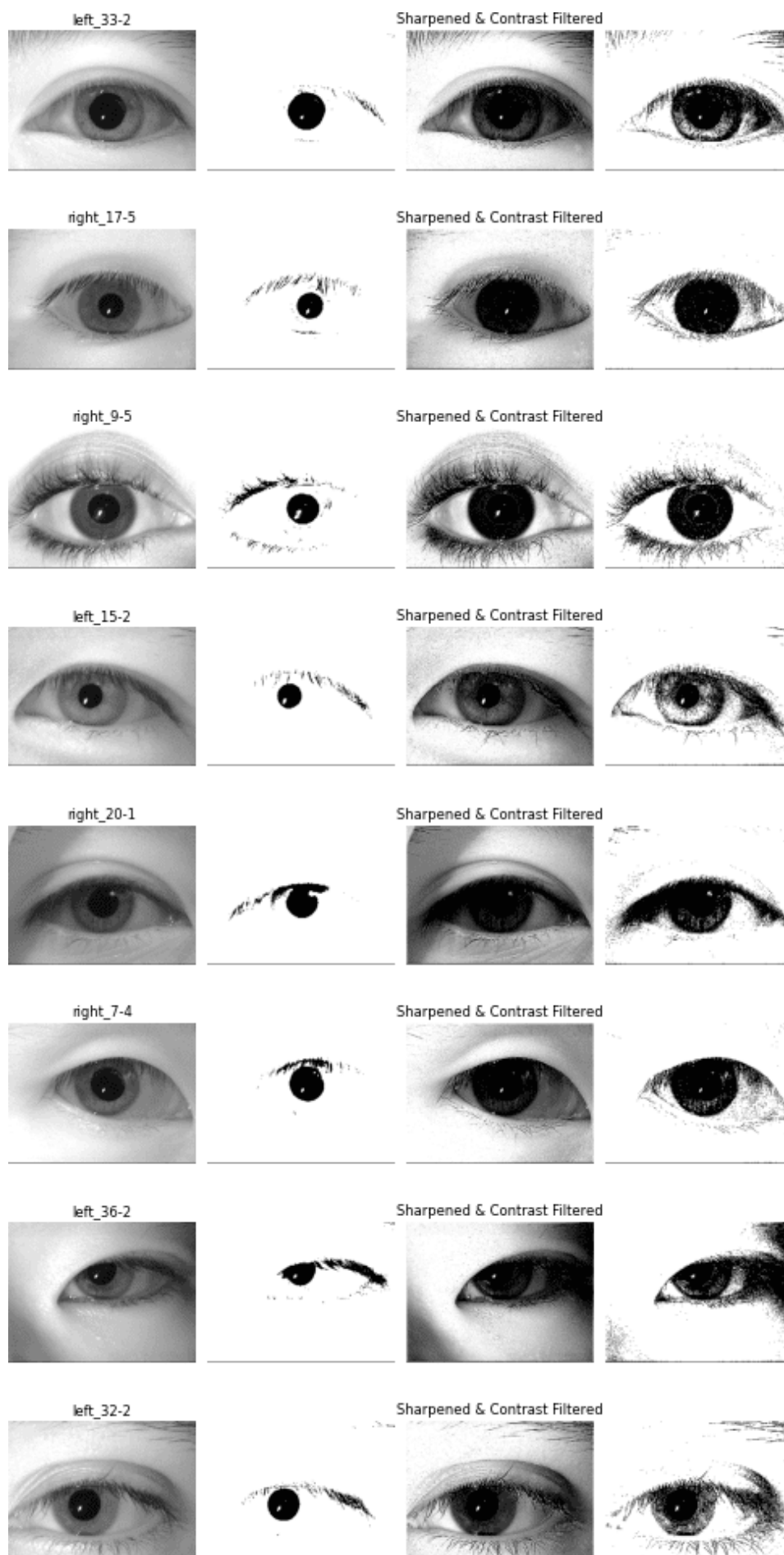


Figure 6: Zastosowanie filtrów poprawiających kontrast oraz wyostrenie obrazu dla lepszej segmentacji źrenicy.

Następnie, dla zsegmentowanych binarnych obrazów źrenicy, wykonano projekcje pionowe i poziome, co pozwala na wyznaczenie środka źrenicy oraz jej przybliżonego promienia. Na ilustracji Figure 7 przedstawiono wyniki projekcji wraz z oznaczeniem centrum oraz obszaru źrenicy.

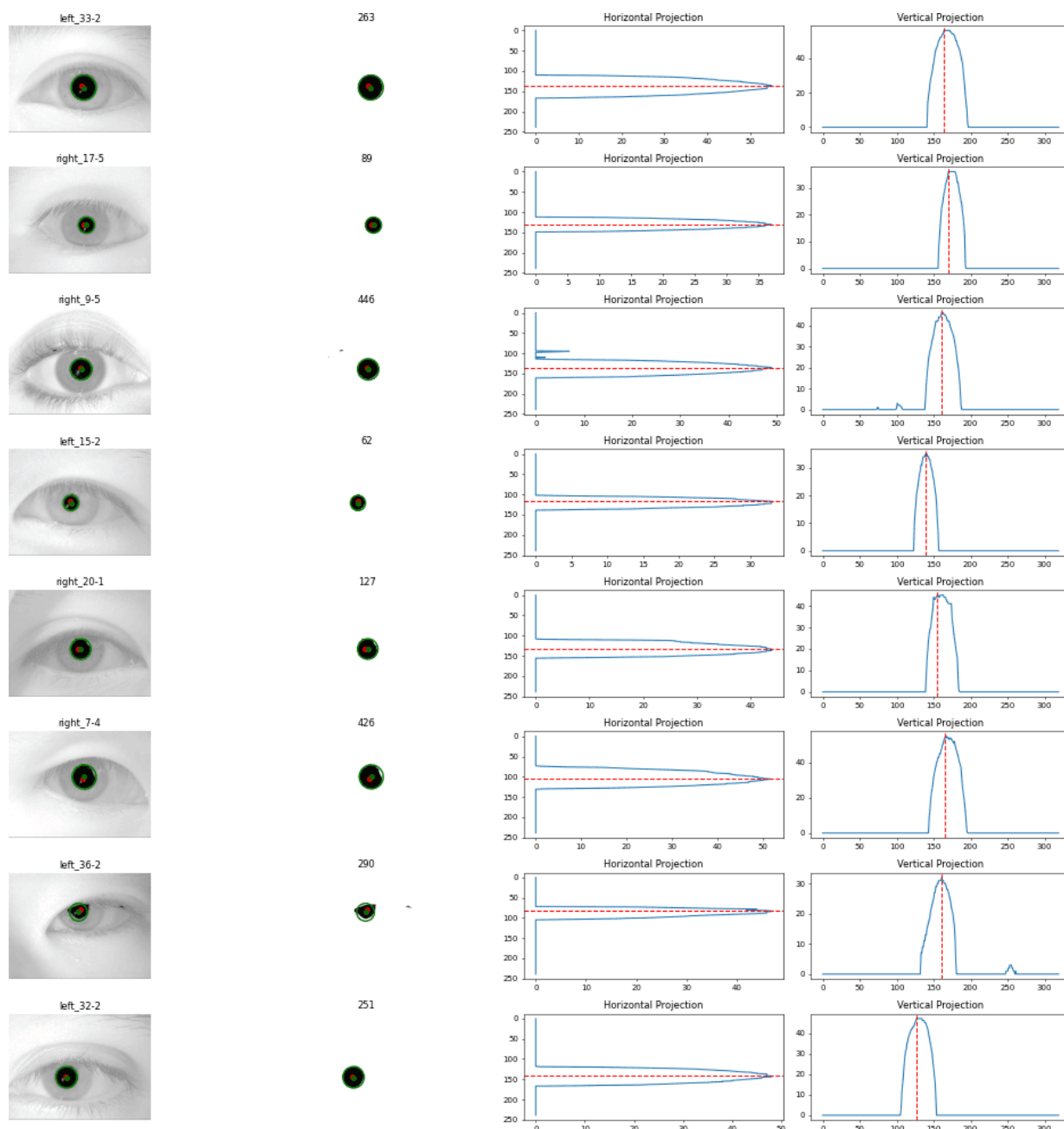
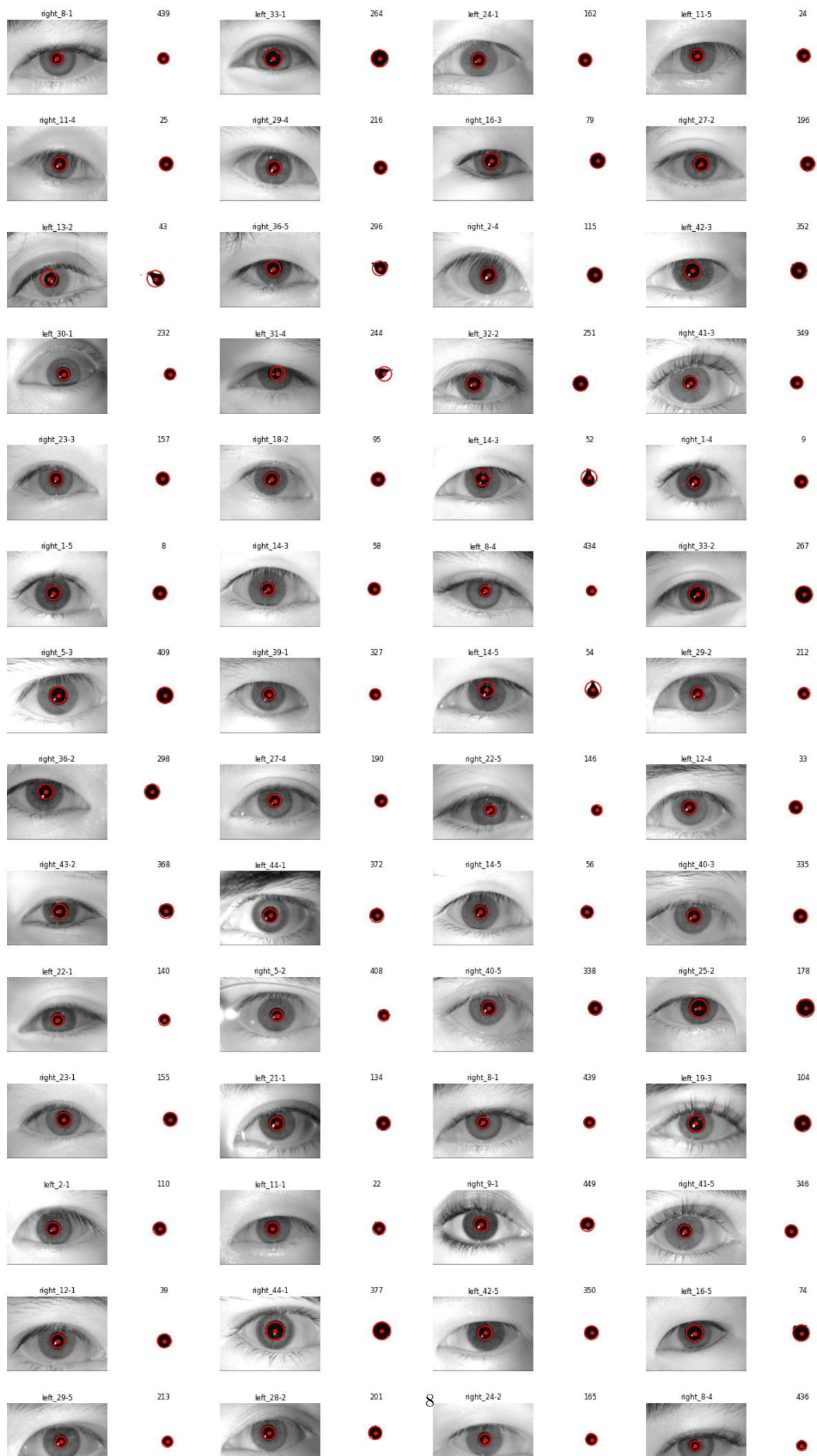


Figure 7: Wizualizacja projekcji poziomych i pionowych dla określenia środka i promienia źrenicy.

Ostateczny rezultat detekcji źrenicy, wraz z naniesionym środkiem i promieniem, zaprezentowano na ilustracji Figure 8. Rysowane okręgi reprezentują wykrytą granicę źrenicy dla różnych obrazów w zbiorze danych. W niektórych przypadkach granice źrenic mogą wydawać się trochę mniejsze od rzeczywistych granic, co wynika z zastosowania operacji morfologicznych. Eksperymenty niezawarte w tym sprawozdaniu wykazały jednak wyższość takiego rozwiązania nad rozwiązaniami dokładnie wyznaczającymi granicę dla zbioru danych "MMU Iris Dataset", ponieważ zdjęcia w nim zawarte mają w wielu przypadkach wyraźne refleksy świetlne zaburzające powodującą błędy w dokładnej detekcji.



2.3 Detekcja granic tęczówki

Po wyznaczeniu granic źrenicy kolejnym etapem jest detekcja granic tęczówki. W tym celu wykorzystano kombinację filtrów: wygładzającego (średniego), kontrastowego oraz filtru Sobela, który umożliwia uwidocznienie krawędzi na obrazie. Przykładowe przekształcenia oraz wyniki binarnej segmentacji obrazu po filtracji przedstawiono na ilustracji Figure 9.

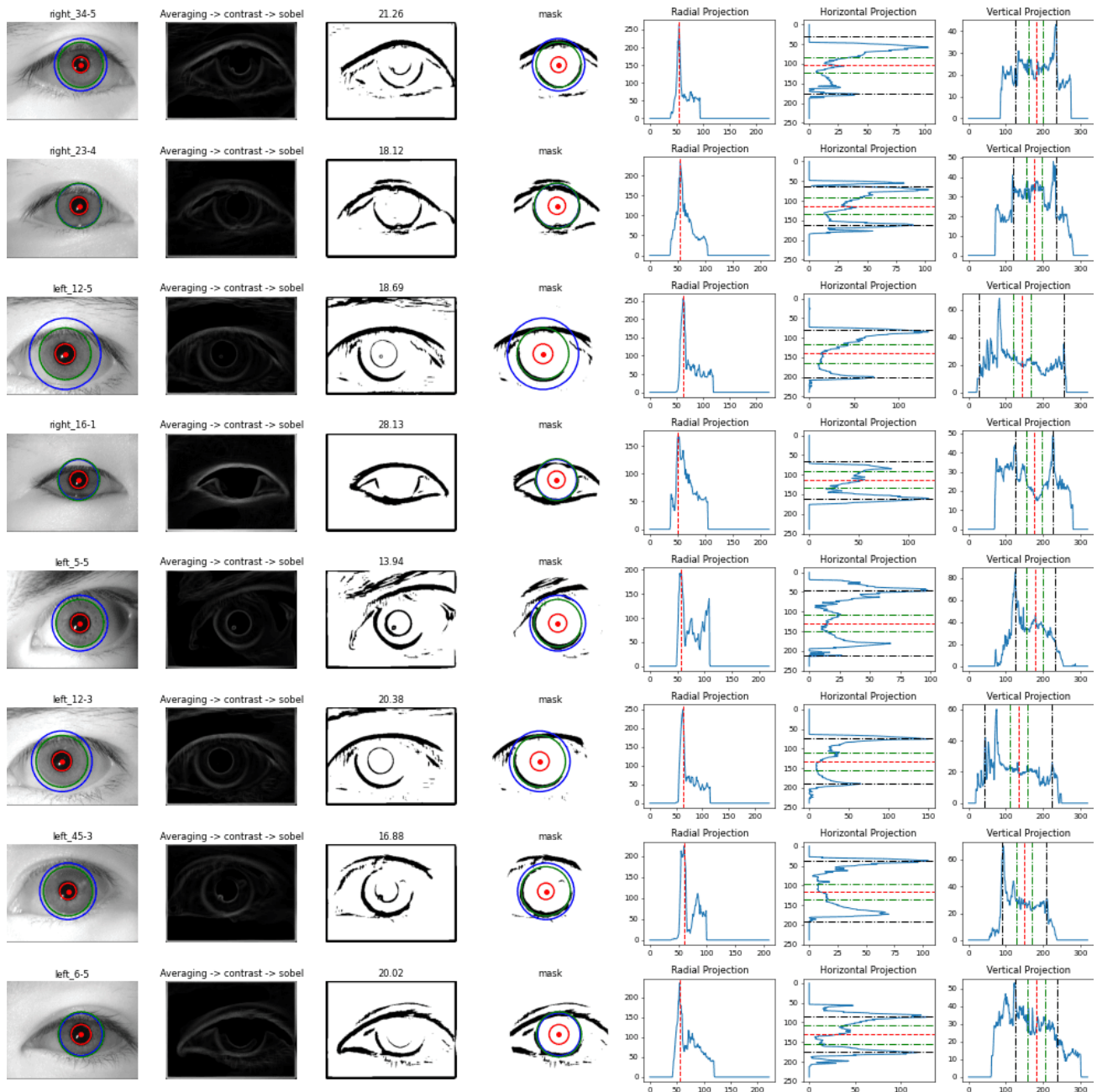


Figure 9: Etapy detekcji tęczówki: przekształcenie obrazu, segmentacja, maskowanie oraz wyznaczenie promienia tęczówki na podstawie trzech projekcji.

Po przetworzeniu obrazu zastosowano binarny próg względem średniej wartości pikseli, a następnie maskowanie zbędnych obszarów: przy brzegach kadru oraz wokół źrenicy. Dzięki temu możliwe było ograniczenie wpływu tła oraz refleksów świetlnych.

Kolejnym krokiem była analiza trzech typów projekcji: poziomej, pionowej oraz radialnej. Projekcja radialna obliczana względem środka źrenicy pozwoliła oszacować promień tęczówki w najbardziej bezpośredni sposób. Jednocześnie promienie wyznaczone z projekcji poziomej i pionowej pozwalały na korektę i uśrednienie wartości. Końcowy promień tęczówki określono jako średnią z promieni uzyskanych metodą projekcyjną (pionową i poziomą).

Na rysunkach zaznaczono granice zarówno źrenicy (kolor czerwony), jak i tęczówki – promień obliczony z projekcji radialnej (zielony) oraz średni promień z analiz poziomej i pionowej (niebieski).

Dla wizualizacji końcowego efektu detekcji, przygotowano ilustrację Figure 10, gdzie widoczne są zarówno oryginalne obrazy tęczy, jak i ich binarne odpowiedniki z zaznaczonymi granicami.

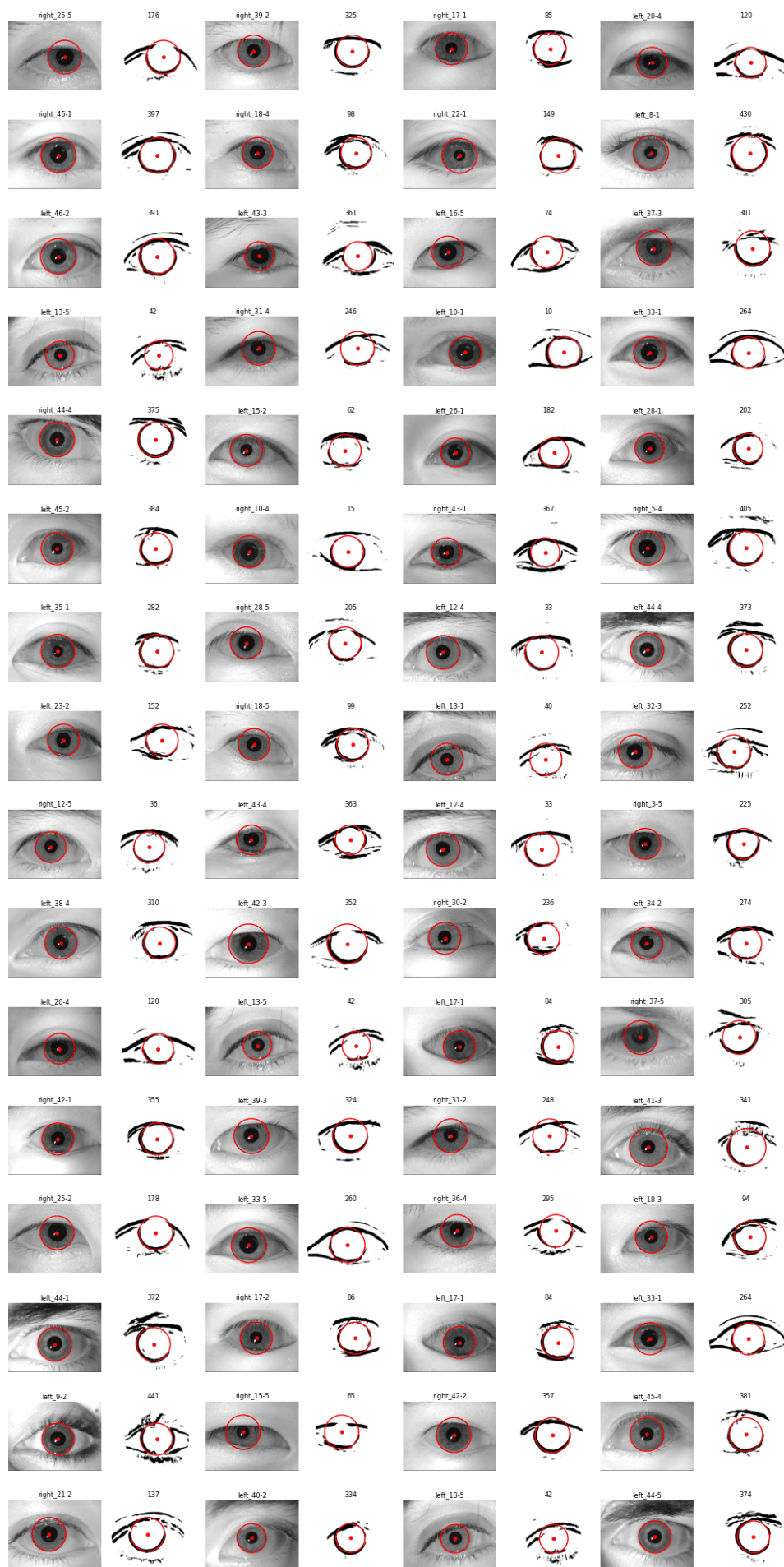


Figure 10: Ostateczne wykrycie granic tęczówek w zestawie danych. Pokazano oryginalne obrazy oraz odpowiadające im segmentacje z zaznaczoną granicą tęczówki.

2.4 Ekstrakcja tęczówki

Po detekcji granic źrenicy i tęczówki kolejnym krokiem jest wyizolowanie samego obszaru tęczówki w celu dalszej analizy. Proces ten polega na wycięciu pierścienia znajdującego się pomiędzy granicami źrenicy i tęczówki. Przykładowy przebieg tej operacji przedstawiono na ilustracji Figure 11.

Na każdej z wizualizacji zaprezentowano:

- oryginalny obraz oka wraz z zaznaczonymi granicami źrenicy (kolor zielony) i tęczówki (kolor czerwony),
- obraz binarny z obszarem tęczówki,
- obraz binarny z obszarem źrenicy,
- oraz wyodrębniony pierścień tęczówki przekształcony do formy prostokątnej (*unwrapped annular segment*).

Przekształcenie polega na rozprostowaniu pierścienia tęczówki na obraz prostokątny – proces ten ułatwia późniejsze etapy analizy i porównywania struktur tęczówek.

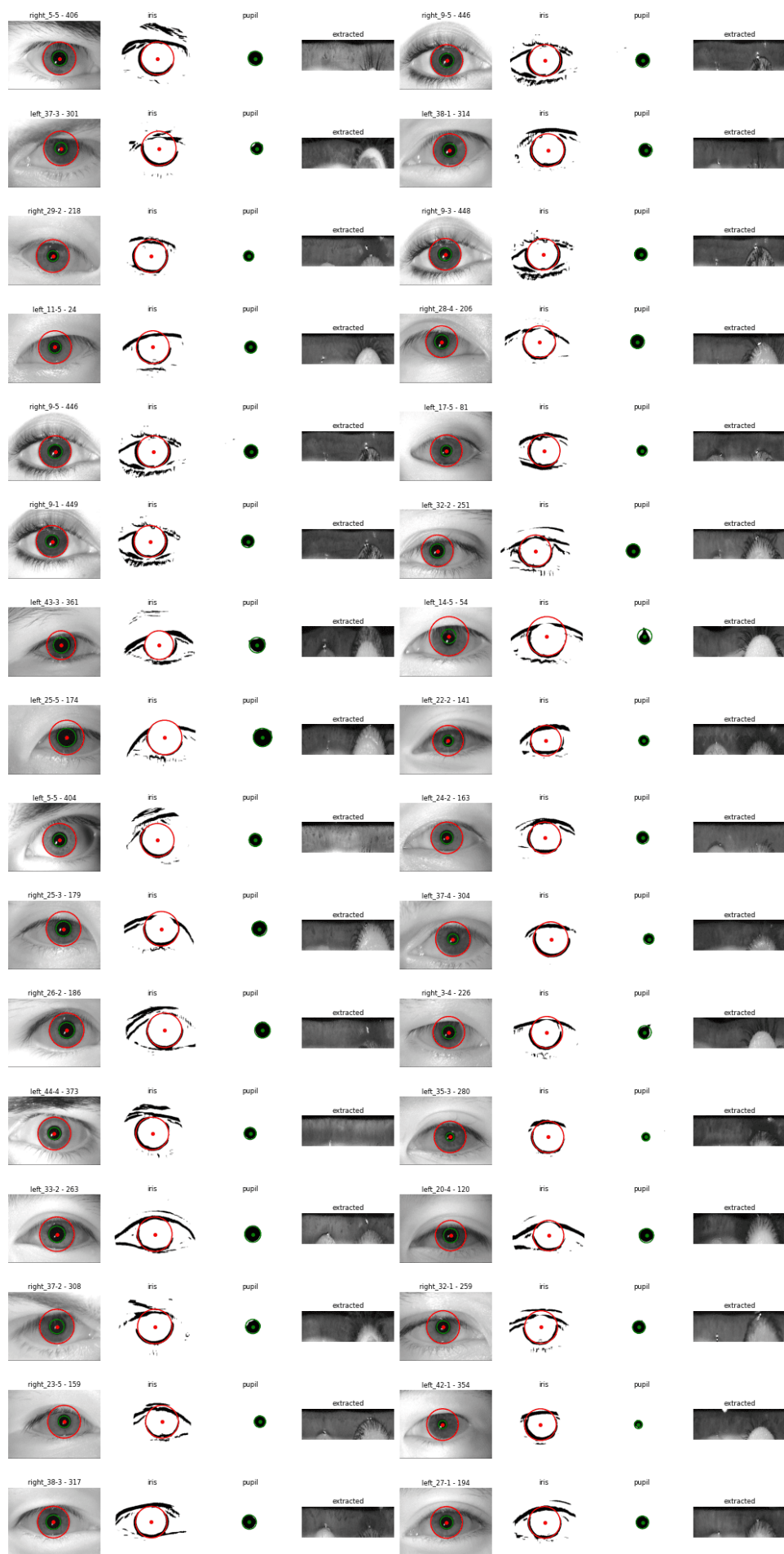


Figure 11: Wizualizacja procesu wycinania tęczówki. Widoczne granice źrenicy i tęczówki, ich maski binarne oraz wyodrębniona tęczówka w postaci rozprostowanego obrazu.

Przekształcone obrazy tęczówek mogą być bezpośrednio wykorzystane do ekstrakcji cech i porównań biometrycznych, takich jak analiza tekstury czy kodowanie wzorców.

3 Algorytm Daugmana

Obróbka wstępna

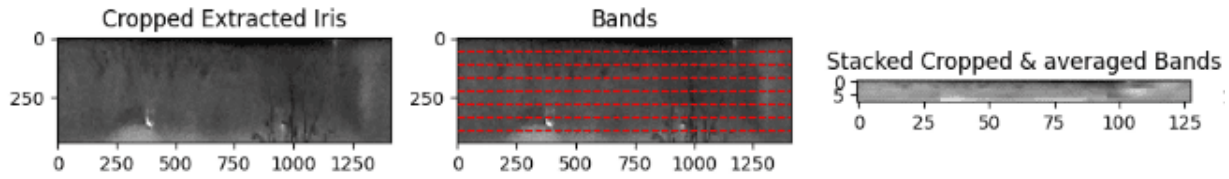


Figure 12: Obróbka wstępna

Aby przygotować obraz do dalszej analizy i segmentacji, proces wstępnej obróbki podzielono na trzy zasadnicze etapy:

1. Przycięcie (crop)

Na wejściu do algorytmu wykorzystujemy obraz o oryginalnych wymiarach 480×1410 pikseli. W celu uniknięcia późniejszej interpolacji przy podziale na równe segmenty, od góry obrazu odcinamy 40 wierszy, uzyskując końcowy rozmiar 470×1410 px. Taki zasadniczy wymiar wysokości (440 px) jest wielokrotnością ośmiu, co pozwala na precyzyjny podział bez utraty informacji o obszarze źrenicy.

2. Podział na pasy i wstępne przycięcie kątowe

Przygotowany obraz dzielimy na 8 poziomych pasów o jednakowej wysokości 55 pikseli ($440 \div 8 = 55$). Następnie, na każdym z pasów, usuwamy fragmenty odpowiadające określonym kątom, zaokrąglając tak, aby szerokość pozostałej części była wielokrotnością 128 pikseli:

- *Pasy 0–3*: usuwamy łącznie ok. 30° obrazu, tak aby pozostała szerokość wynosiła 1280 px.
- *Pasy 4–5*: usuwamy po ok. 67° zarówno od góry, jak i od dołu, uzyskując szerokość 768 px.
- *Pasy 6–7*: analogicznie usuwamy łącznie ok. 134° (po ok. 67° od góry i od dołu), co daje szerokość 640 px.

3. Uśrednienie radialne

Z każdej z uzyskanych części obliczamy średnią wartość pikseli w kierunku pionowym, sprowadzając każdy pas do wysokości 1 px oraz szerokości 128 px. W efekcie końcowym otrzymujemy osiem pasów o wymiarach 1×128 px, gotowych do dalszej analizy.

Generowanie kodów tęczówki

Eksperymentalne wyznaczenie częstotliwości

Istotnym parametrem używanym do identyfikacji osoby, a nie podanym w publikacjach Daugmana, jest częstotliwość. Jej znaczenie demonstrują rysunki Figure 13 oraz Figure 14. Na pierwszym z nich widać istotny wzrost kodowanej informacji wraz ze wzrostem częstotliwości. Na drugim zaś – jak zmienia się średnia odległość Hamminga wraz ze zmianą częstotliwości. Pomiar ten wykonywany jest poprzez porównanie pierwszego zdjęcia z grupy pięciu zdjęć jednej osoby (kodowane) z pozostałymi oraz z innymi osobami. Dla częstotliwości 0.001 różnice są znikome, natomiast przy wartościach wzrastających do około 0.5 zauważalne stają się istotne rozbieżności. Na tej podstawie przyjęto wartość 0.5 do dalszego przetwarzania.

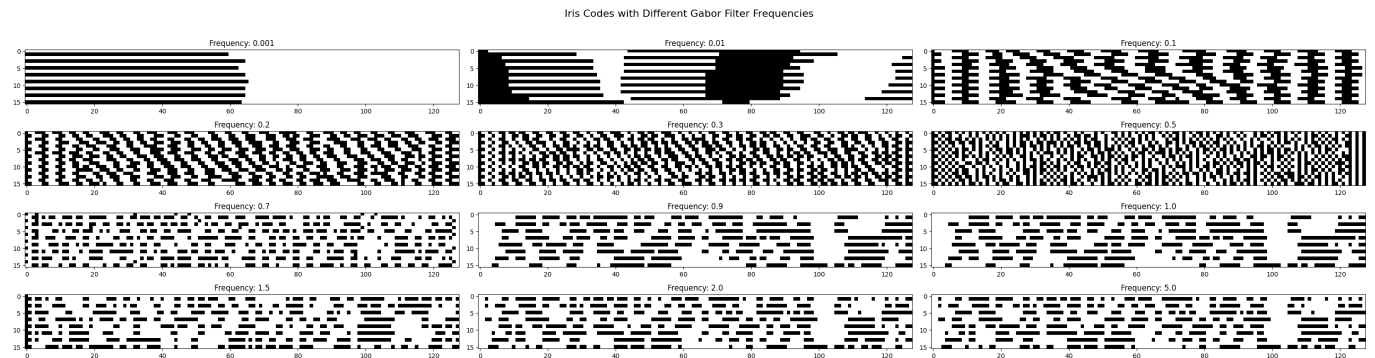


Figure 13: Wygląd kodowania na podstawie częstotliwości

Hamming Distance Between Iris Codes with Different Frequencies

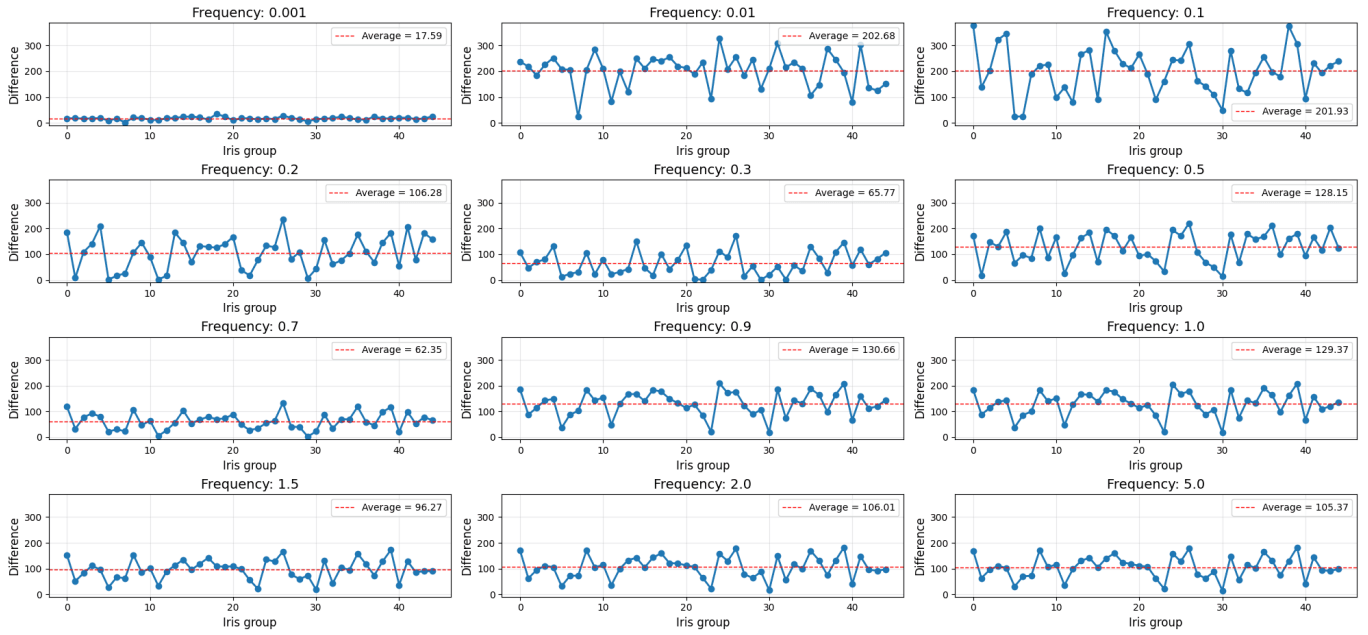


Figure 14: Odległość Hamminga między tęczówkami różnych osób

4 Demo

W tej części przedstawiono wyniki eksperymentów, które potwierdzają, że tęczówka jest skutecznym narzędziem do identyfikacji biometrycznej.

Ta sama osoba, oko lewe i prawe

Na poniższym zdjęciu przedstawiono kody tęczówki tej samej osoby, lecz dla dwóch różnych oczu. Zgodnie z hipotezą, wzorce tęczówek lewego i prawego oka powinny się różnić – co wyraźnie widać na obrazie. Białe pola wskazują miejsca rozbieżności pomiędzy kodami, które stanowią około 50

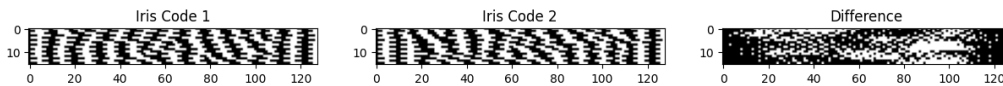


Figure 15: Oko lewe i prawe tej samej osoby

Ta sama osoba, to samo oko

Kolejnym przykładem jest porównanie kodów wygenerowanych dla tego samego oka na podstawie dwóch różnych zdjęć. Pomimo różnic warunków rejestracji, wygenerowane kody pozostają do siebie bardzo podobne, co umożliwia skuteczną identyfikację osoby.

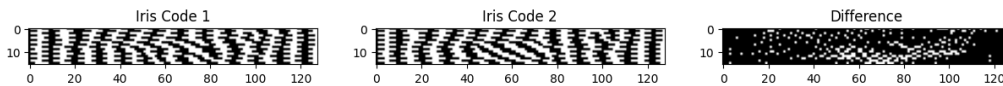


Figure 16: Różne zdjęcia tego samego oka

Różne osoby

Porównanie kodów tęczówek różnych osób wykazuje wyraźne różnice. Na rysunkach Figure 17 oraz Figure 18 widać liczne białe pola, świadczące o braku zgodności pomiędzy kodami, co umożliwia jednoznaczną identyfikację.

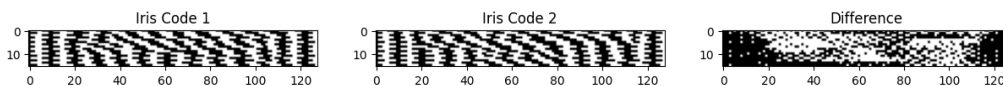
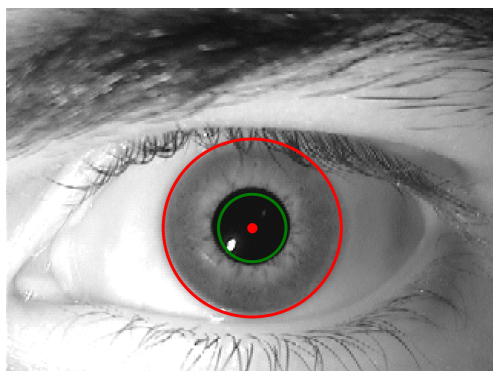
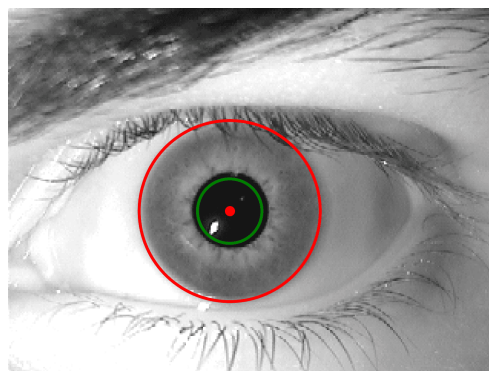


Figure 17: Kody tęczówki różnych osób



(a) Zdjęcie 44-1



(b) Zdjęcie 44-2

Figure 18: To samo oko, ale różne zdjęcia

Performance tests

Przeprowadzono prosty test wydajnościowy, z którego wynika, że zakodowanie rozprostowanego obrazu tęczówki zajmuje ok. 0.0039 sekundy, co daje około 253 kodów na sekundę. Samo uzyskanie tego kodu zajmuje średnio 0.0012 sekundy (810 kodów na sekundę). Mimo braku optymalizacji, wydajność systemu jest bardzo wysoka, dając całościową prędkość systemu (licząc wczytywanie zdjęcia do ramy) rzędu 253 kodów na sekundę. Z pewnością dodatkowe optymalizacje mogą poprawić ten wynik jeszcze bardziej.

5 Podsumowanie

W projekcie zrealizowano kompletny łańcuch przetwarzania obrazu tęczówki, zgodnie z klasycznym podejściem Johna Daugmana – od surowych zdjęć oka do binarnego *kodu tęczówki* stosowanego w biometrii.

1. **Zbiór danych:** Wykorzystano publiczny zbiór *naureenmohammad/mmu-iris-dataset* (45 osób \times 2 oczy \times 5 ujęć). Pomimo wyzwań związanych z jakością zdjęć, uzyskano stabilną segmentację.
2. **Segmentacja oka:**
 - *Żrenica:* stosowano progowanie po filtrach kontrastu i wyostrażania; projekcje poziome i pionowe ustalają środek i promień.
 - *Tęczówka:* filtr średni, wzmocnienie kontrastu, filtr Sobela; maskowanie tła i żrenicy; promień wyznaczany z projekcji radialnej, poziomej i pionowej, a następnie uśredniany.
3. **Ekstrakcja pierścienia tęczówki:** Wycięto annulus między żrenicą a brzegiem tęczówki i „rozprostowano” go do formy prostokątnej.
4. **Algorytm Daugmana:** Po przycięciu obrazu, podziale na osiem pasów i uśrednieniu radialnym, eksperymentalnie ustalono optymalną częstotliwość filtra Gabora na poziomie około 0.5, maksymalizując różnicę Hamminga.
5. **Eksperymenty weryfikacyjne:**
 - *Lewe vs prawe oko tej samej osoby:* około 50
 - *To samo oko, różne zdjęcia:* niskie odległości Hamminga potwierdzają stabilność kodów.
 - *Różne osoby:* kody znacząco się różnią, umożliwiając jednoznaczną identyfikację.
 - *Wydajność:* czas generacji kodu wynosi ok. 0.0023 s, co daje ok. 437 kodów/s.

Wniosek: Zaproponowana ścieżka przetwarzania obrazu tęczówki spełnia wymagania systemów biometrycznych, zapewniając wysoką rozróżnialność międzyosobniczą, niską zmienność wewnątrzosobniczą oraz dużą wydajność operacyjną.