Fingerprints processing

Bartosz Jezierski

18 maja 2025

Spis treści

1 Wprowadzenie

Celem projektu było porównanie dwóch algorytmów ścieniania - klasycznej szkieletyzacji morfologicznej i KMM w odniesieniu do odcisków palców. Następnie z uzyskanych szkieletów należało wyznaczyć zakończenia i bifurkacje linii papilarnych w celu dalszej (niewymaganej już) weryfikacji/identyfikacji odcisków.
W projekcie tym korzystam z już wcześniej zaimplementowanych operacji na zdjęciach z poprzednich dwóch projektów. Wszystkie potrzebne funkcje zostały zaimportowane w notebook'u w sekcji *Funkcje z poprzedniego proj.*
Z kolei wszystkie nowe funkcje potrzebne do uzyskania wszystkich wyników z tego projektu są zaimlpementowanie w sekcji *Nowe funkcje* w notebook'u.
Funkcje przedstawione na wykładzie starałem się w miarę możliwości testować na przykładach z prezentacji w celu weryfikacji poprawności działania.

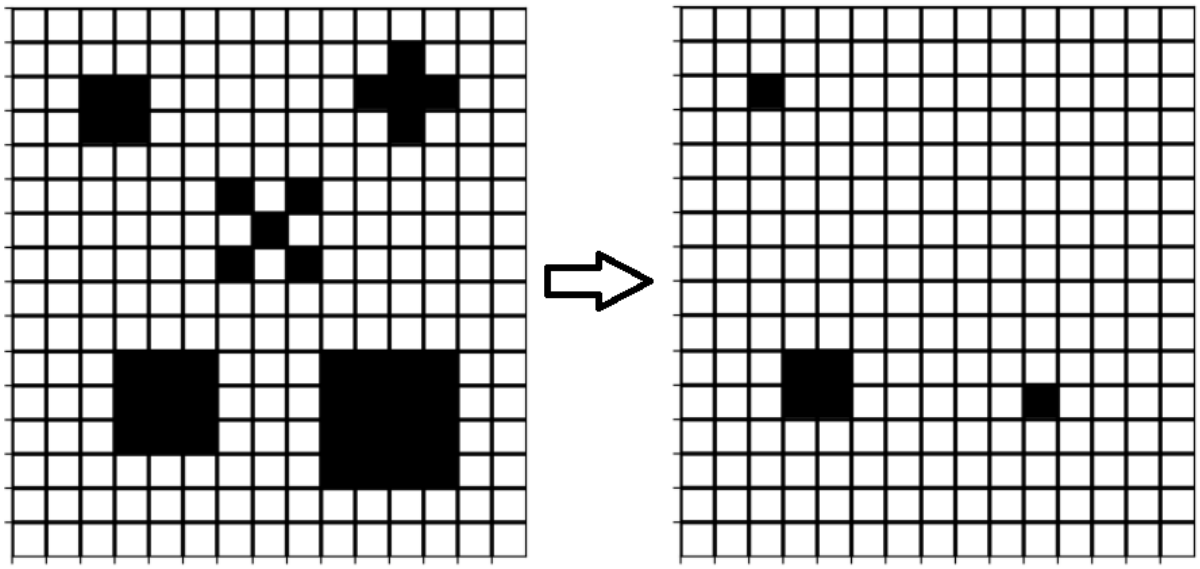
2 Klasyczna szkieletyzacja

2.1 Hit and Miss

```
1 def hit_and_miss(image, se_hit, rep_hit, se_miss, rep_miss):
2     """
3     Operacja hit_and_miss dla binarnego obrazu
4     """
5     erosion_obj = erosion(image, se_hit, rep_hit)
6     erosion_bg = erosion(negation(image), se_miss, rep_miss)
7 
8     return np.logical_or(erosion_obj, erosion_bg)
```

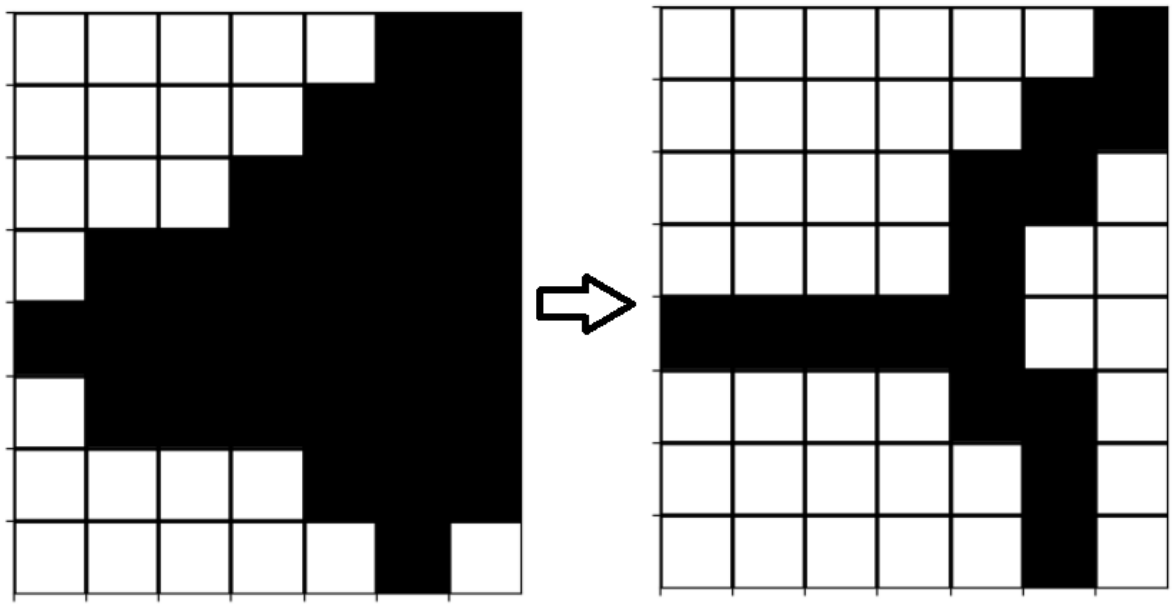
Listing 1: Implementacja Hit and Miss

Warto zauważyć (a raczej przypomnieć), że u nas obiekt reprezentuje pixel o wartości 0, natomiast tło reprezentuje pixel o wartości 1. Stąd wszystkie funkcje implementują wewnętrznie operacje odwrotne do tych przedstawionych na wykładzie (np. or zamiast and powyżej).



Rysunek 1: Efekt działania operacji Hit And Miss (test czy działa tak jak wersja wykładowa).

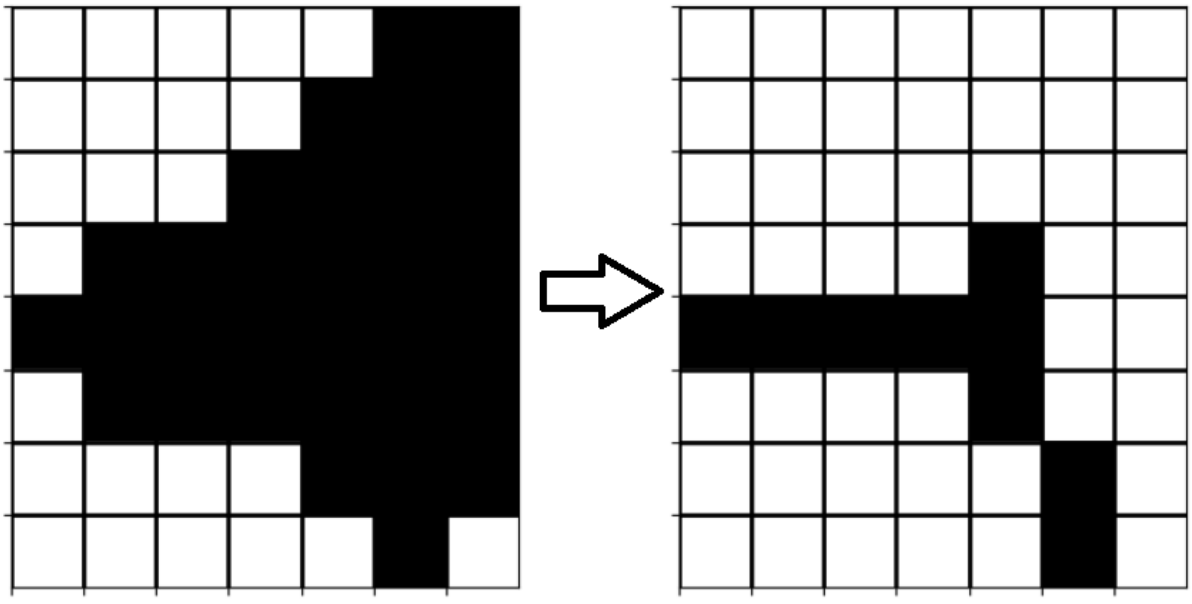
2.2 Test działania szkieletyzacji 4 path z prezentacji



Rysunek 2: Efekt działania zaimplementowanej operacji szkieletyzacji 4_path (test czy działa tak jak wersja wykładowa).

W notebook'u dostępna jest jeszcze wersja ukazująca kolejne kroki algorytmu.

2.3 Test działania szkieletyzacji 8 path z prezentacji



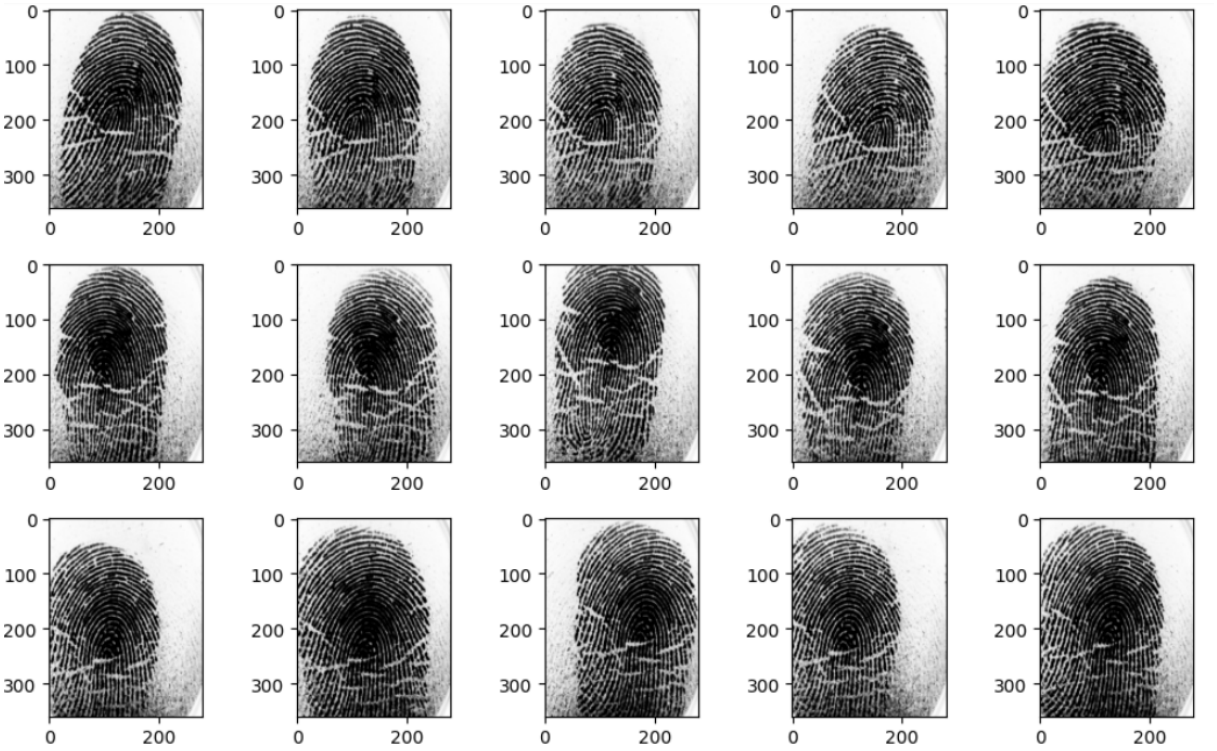
Rysunek 3: Efekt działania zaimplementowanej operacji szkieletyzacji 8_path.

Tutaj miałem problem ze zrozumieniem jak powinny wyglądać poszczególne elementy strukturalne (dlatego wyszedł niepoprawny wynik). Stąd też uznałem, że dalej będę testował tylko szkieletyzację 4_path jako, że była lepiej opisana na prezentacji.

3 Działanie klasycznej szkieletyzacji na odciskach

3.1 Zbiór testowy

Miałem do dyspozycji 50 odcisków palców z moich rąk (po 5 na każdy palec). Okazuje się, że linie papilarne moich palców są bardzo poprzecinane (nie wiem czy jest to kwestia suchych palców, czy po prostu tak mam). Przedstawiam próbkę odcisków poniżej dla pierwszych trzech palców.



Rysunek 4: Pierwsze 3 palce, widać bardzo dużo przecięć.

3.2 Przykładowe działanie algorytmu bez preprocessingu zdjęcia.

Jak widać tworzą się złe połączenia (takie których nie ma) między sąsiednimi liniami papilarnymi tworząc siatkę "kwadratów"w centrum co zdecydowanie nie sprzyja znajdowaniu poprawnych bi-

furkacji. Ponadto dla dalszych od centrum linii ich połączenia są bardzo poprzerywane. Trzeba poprawić zdjęcie wejściowe. Na wchodzącym do thinningu obrazie linie są strasznie poprzerywane.

4 Poprawa wyniku

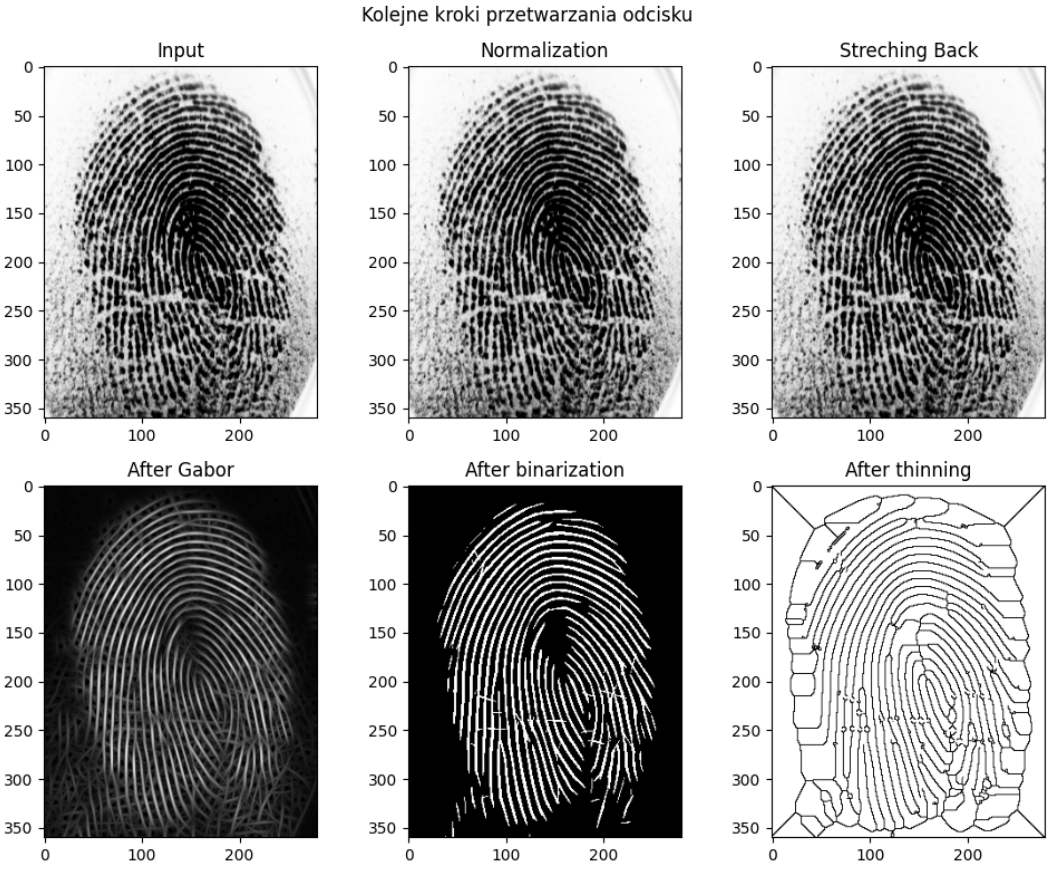
W celu ulepszenia szkieletu wprowadziłem wstępny preprocessing zdjęcia za pomocą poniższych metod.

4.1 Normalizacja wejścia i filtr Gabora

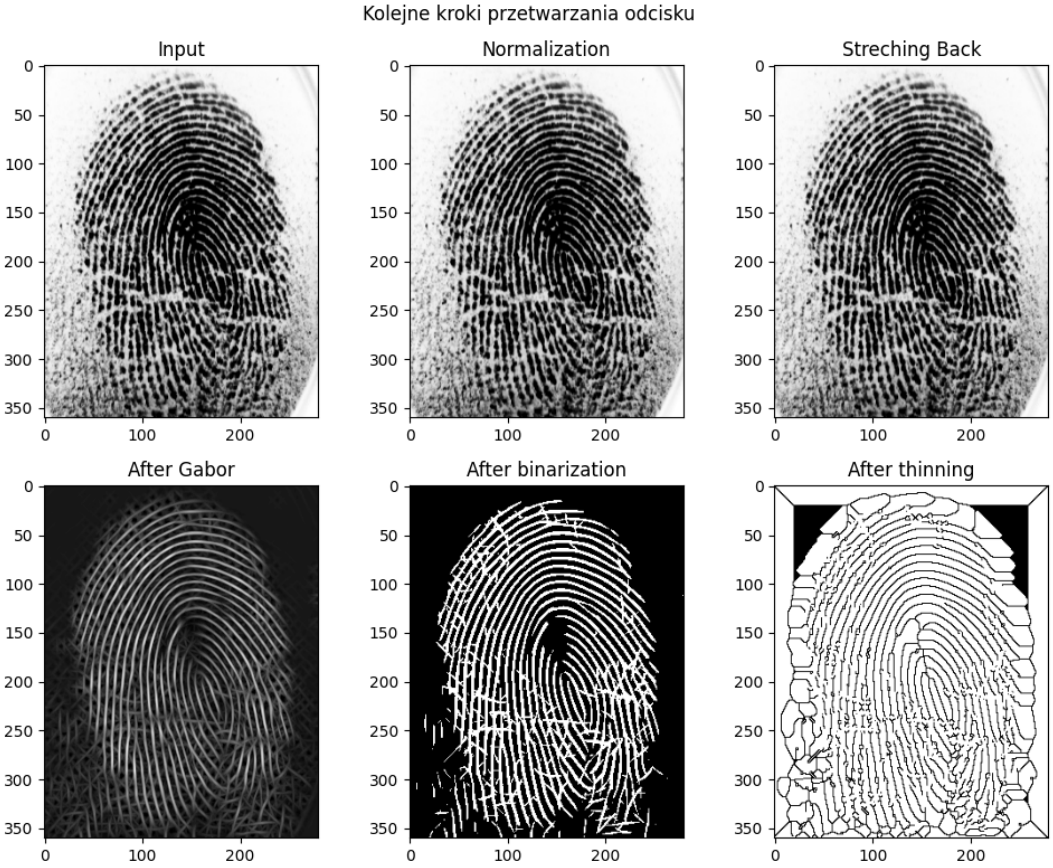
Aby ograniczyć wpływ złego zdjęcia postanowiłem najpierw normalizować zdjęcie (przenosić je na przestrzeń 0-255, żeby wykorzystywać pełny zakres możliwych wartości pixeli). Następnie zdjęcie było ulepszane przez filtr Gabora, a konkretniej pewnego rodzaju uśrednienie zdjęcia wieloma filtrami Gabora dla różnych kątów i częstotliwości (funkcja ta znajduje się w sekcji *Nowe funkcje* w podsekcji *Implementacja filtr Gabora* w notebook'u). Za pomocą odpowiedniego doboru parametór mogłem dzięki temu:

- wypełnić praktycznie całkowicie przerwania między liniami kosztem wypełnionego niepoprawnie środka odcisku palca, bądź
- wypełnić częściowo nieciągłości linii papilarnch zostawiając naturalne przecięcia na palcu oraz środek w bardziej poprawnej wersji.

Poniżej przedstawiam dwie możliwości.



Rysunek 5: Możliwość pierwsza

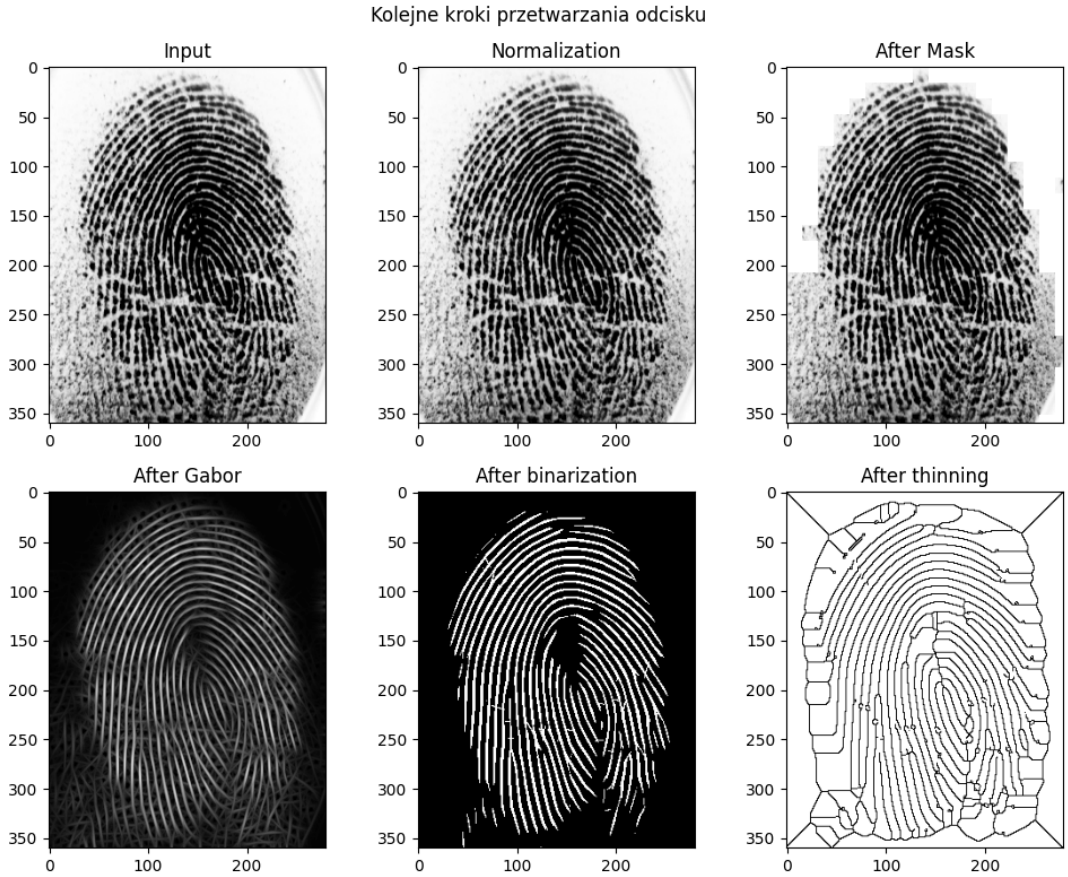


Rysunek 6: Możliwość druga

Osobiście skłaniam się bardziej ku opcji nr 1, jednak końcowo udało mi się zachować balans między tymi twoma możliwościami.

4.2 Maska

Dodałem również formę eliminacji szumu poprzez zastosowanie dopasowującej się maski do odcisku palca (starając się zachować tylko odcisk). Maska tworzy się poprzez podzielenie zdjęcia na kwadraty o równej ilości pixeli, by następnie w każdym z nich policzyć wariancję. Jeśli sprawdzany kwadrat przekracza podany w parametrach próg wariancji to jest on uwzględniony w dalszym procesie.



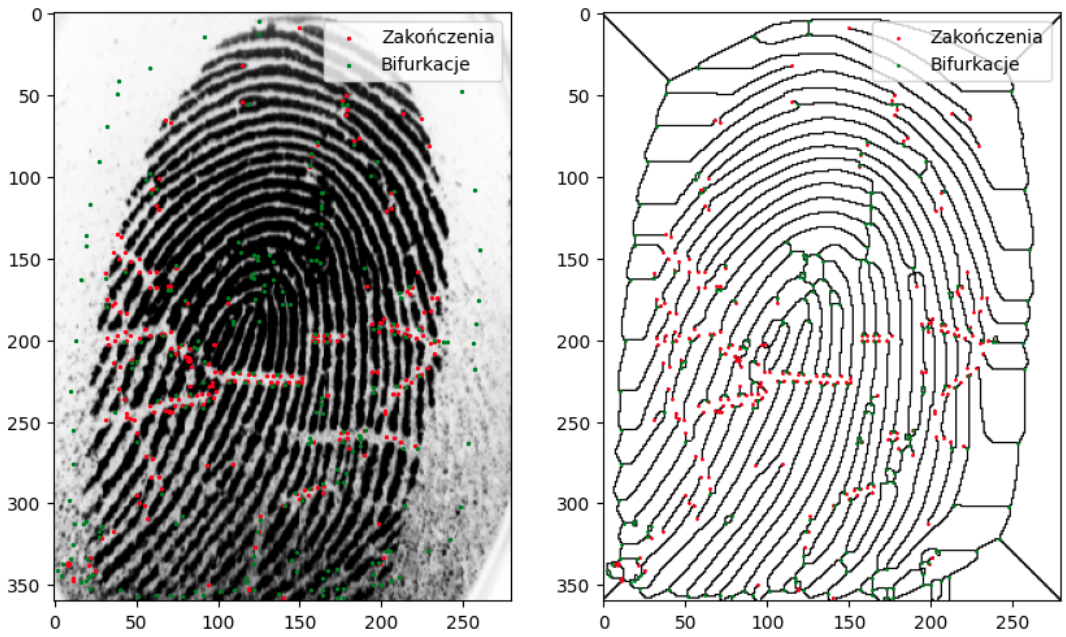
Rysunek 7: Dodanie maski do preprocessingu w celu eliminacji szumu.

4.3 Próba dalszych usunięć pozostałych mikro przecięć

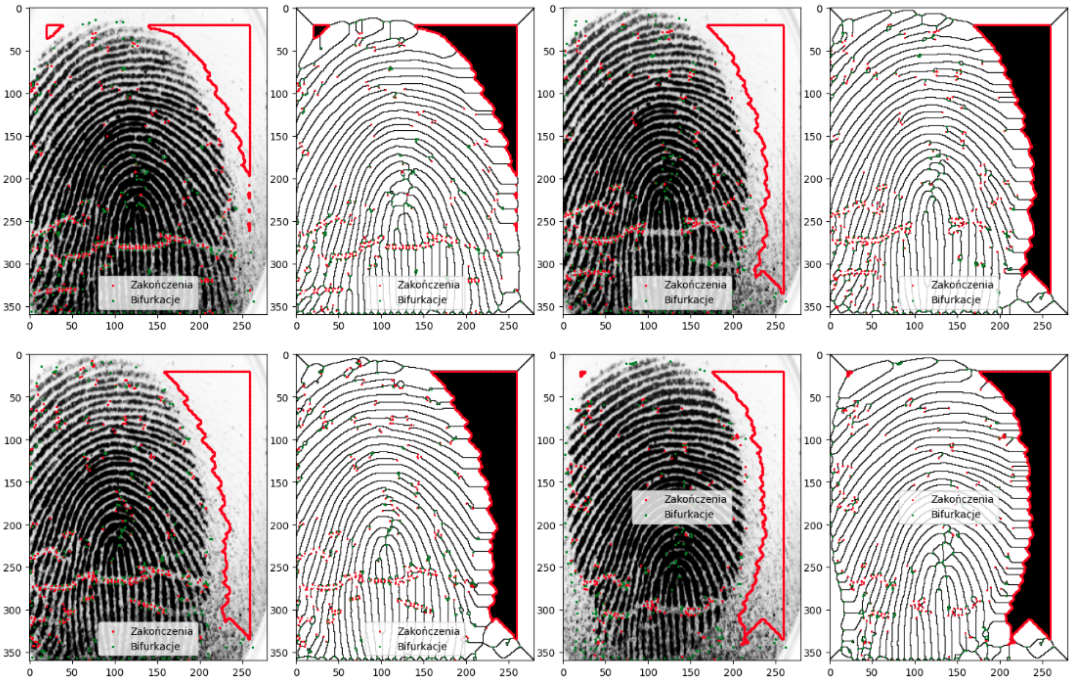
Próbowałem dalej usunąć pozostałe mikro przecięcia za pomocą morfologicznych operacji zamknięcia, jednak z marnym skutkiem. Doprowadzało to za każdym razem do albo usunięcia co najwyżej paru przecięć kosztem połączenia się czasem sąsiednich linii papilarnych, bądź usunięcia przecięć kosztem całkowitego zniszczenia (połączenia się) struktury linii papilarnych. Stąd nie zamieszam tu już żadnych wyników dotyczących tego podejścia.

4.4 Lokalizacja minuncji

Teraz pozostało tylko znaleźć zakończenia i bifurkacje. Znajdowanie rdzeni i delt zostało końcowo usunięte z wymagań dotyczących projektu, jako, że prowadzący stwierdził, że nie było to poruszane na wykładzie. Jak widać, na kolejnym rysunku widać to czego można było się spodziewać. Algorytm znajduje wiele nieistniejących bifurkacji w środku odcisku. Poza tym wydaje się na oko, że poza środkiem odcisku algorytm znajduje poprawnie wymagane minuncje.



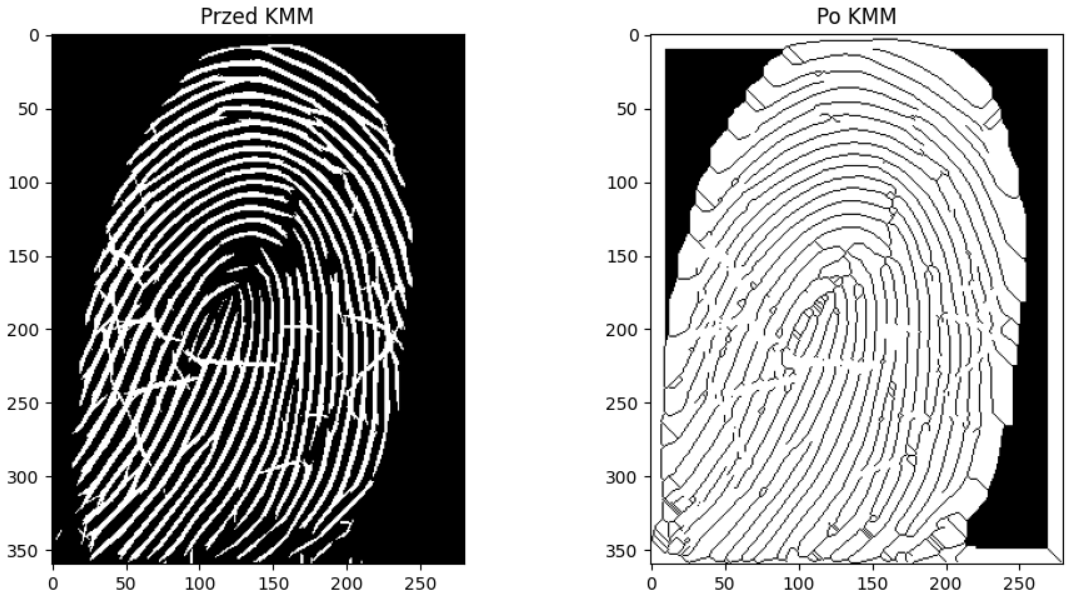
Rysunek 8: Znalezione zakończenia i bifurkacje.



Rysunek 9: Znalezione zakończenia i bifurkacje.

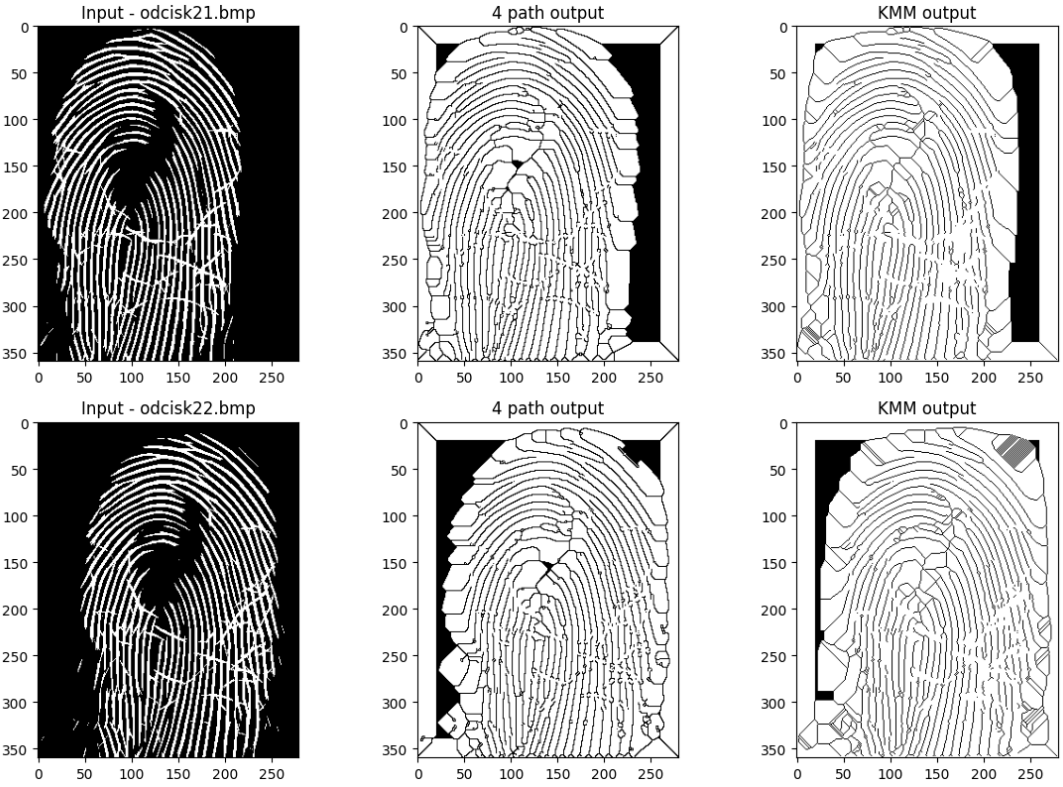
5 Szkieletyzacja za pomocą KMM

Preprocessing do tej szkieletyzacji stosowałem dokładnie taki sam jak do poprzedniego typu szkieletyzacji. W notebook'u dostępna jest wizualizacja z kolejnymi krokami algorytmu, z pokazaniem jakie oznaczenia liczbowe przypisywane są pixelom w danych krokach (*Sekcja Nowe Funkcje -> KMM*).
Poniżej przedstawiam przykład działania na odcisku palca.



Rysunek 10: Przykład resultatu KMM.

5.1 Porównanie klasycznej szkieletyzacji i KMM



Rysunek 11: Porównanie działania KMM i 4_path.

KMM lepiej (przynajmniej moim zdaniem) odwzorowuje zakończenia (podczas gdy 4_path często tworzy z jednego zakończenia podwójne zakończenie, co najbardziej widać na przecięciach). KMM jednak tworzy mnóstwo sztucznych bifurkacji (Wystarczy jeden biały pixel wokół czarnych i tworzy się kreska, jakich mnóstwo jest widocznych na ostatnim wykresie w prawym, górnym rogu obrazu odcisk 22 przetworzonego przez KMM)