POLITECHNIKA WARSZAWSKA

WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH

Rozprawa Doktorska

mgr inż. Kacper Sarnacki

Poprawa jakości zdjęć pisma ręcznego z wykorzystaniem morfologii adaptacyjnej.

Promotor

prof. dr hab. inż. Khalid Saeed

WARSZAWA 2021

Streszczenie

Niniejsza praca przedstawia algorytm usprawniający jakość zdjęć pisma ręcznego z wykorzystaniem morfologii adaptacyjnej. Celem jej powstania jest udowodnienie tezy, że istnieje uniwersalne podejście do poprawiania jakości binaryzacji obrazów zawierających obiekty o liniowej strukturze, które da lepsze wyniki w porównaniu do binaryzacji. Niewątpliwą zaletą prezentowanego podejścia jest modularność. Algorytm współpracuje z dowolną metodą binaryzacji. Sposób wyznaczania pola kierunkowego może być także wybrany przez użytkownika. Ponadto prezentowany algorytm proponuje stosowanie komponentów opcjonalnych. Pozwalają one uzyskać jeszcze lepszy rezultat, jednak kosztem złożoności obliczeniowej. Przykładem takich rozszerzeń jest zwiększanie próbkowania czy test 4-sąsiedztwa. Elastyczność prezentowanej metody pozwala znaleźć balans pomiędzy jakością zwracanego rozwiązania a wydajnością działania. Wysoką skuteczność algorytm zawdzięcza identyfikacji pola kierunkowego, dzięki czemu możliwe jest przewidzenie kształtu linii czy ewentualnych przecięć. Na tej podstawie można dostosować sposób naprawiania ubytków wewnątrz linii poprzez zastosowanie różnych typów morfologii.

Szczególnie istotną częścią niniejszej pracy jest weryfikacja jakości stworzonego algorytmu. Z racji jego elastyczności, testom zostały poddane różne wersje, składające się z wymiennych komponentów. W celu obiektywnej oceny rozwiązania, zastosowano trzy rodzaje weryfikacji. Użyto testów wizualnych, analitycznych i statystycznych. Operowanie na obrazach przedstawiających pismo pozwala w łatwy sposób weryfikować uzyskane usprawnienia, dzięki powszechnie znanej wiedzy na temat kształtu liter. Test wizualny jest użyteczny także przy uwzględnieniu faktu, że w niektórych sytuacjach poprawianie rezultatów binaryzacji jest ostatnim etapem automatycznego przetwarzania obrazów, zanim tak przetworzone zdjęcie trafi do specjalisty. Kolejnym rodzajem testów jest test analityczny, który opisuje liczbowo jakość algorytmu. Dzięki niemu można porównać dokładność różnych wersji algorytmów. Ta metoda porównuje wyniki z rezultatem idealnym. Dzięki temu można ocenić, czy algorytm poprawiający jakość binaryzacji rzeczywiście polepsza ostateczny rezultat oraz w jakim stopniu różni się od wyniku idealnego. Testy statystyczne udowadniają, że poprawa jakości obrazu poprzez stosowanie prezentowanego algorytmu nie jest przypadkowa i że jest to zmiana istotna statystycznie.

Abstract

This document presents an algorithm to improve the quality of photos with handwritten text with the use of adaptive morphology. The aim is to prove the thesis that there exists an universal approach to improve binarization effects of the images containing objects with a linear structure, which will give better results compared to binarization. One of the biggest advantages of the presented approach is its modularity. The algorithm works with any binarization method. The method for determining a directional field may also be selected by the user. In addition, the presented algorithm proposes using some optional components. They allow you to get an even better result but computational complexity increases with the same time. Examples of such extensions are upsampling and the 4-neighborhood test. The flexibility of the presented method allows to find a balance between the quality of the returned solution and its efficiency. High accuracy of the algorithm colud be achieved thanks to identifying directional field, thanks to which it is possible to predict the shape of the line or potential intersections. Based on this information, it is possible to adapt the way of repairing defects inside the line by applying different types of morphology.

A especially important part of this document is the quality verification of the designed algorithm. Due to its flexibility, various versions were tested, consisting of replaceable components. In order to objectively test the algorithm, three types of verification were used. Visual, analytical and statistical tests were executed. Operating on images presenting letters is an easy way to verify and compare the obtained results, thanks to the fact that the shape of the letters is a common knowledge. The visual test is also useful in some situations when preprocessing is the last stage of automatic image processing and then the processed image goes to a specialist. Another type of test, analytical test, quantifies the quality of the algorithm. With the use of this kind of test, it is possible to compare the accuracy of different versions of the algorithms. This method compares the results with the ideal result. Thanks to this, it is possible to assess whether the preprocessing algorithm really improves the final result and how much is missing from the ideal result. Statistical tests prove that the improvement of image quality through the use of the presented algorithm is not accidental and that it is a statistically significant change.

Spis treści

Wstęp

W dzisiejszych czasach mamy do czynienia z rozrostem zapotrzebowania na systemy przetwarzające obrazy. Wraz z rosnącą digitalizacją sektora usługowego, coraz więcej przedsiębiorstw decyduje się na przeniesienie całkowitej komunikacji online. Nierzadko wiąże się to z potrzebą przeniesienia dużych zbiorów dokumentów na serwery sieciowe. Aby móc wykorzystywać te dane w praktyce, każda strona musi zostać przetworzona na tekst. Alternatywą do wykonywania takiej pracy manualnie są systemy automatycznego rozpoznawania tekstu. Takie systemy są coraz częściej wykorzystywane przez banki, wśród których istnieje duże zapotrzebowanie na gromadzenie danych. Te dane są następnie wykorzystywane do uczenia maszynowego systemów przewidujących zachowanie rynku [20, 88]. Podobny trend można zauważyć także w licznych gałęziach rozrywki. Serwisy społecznościowe takie jak Instagram czy TikTok zostały zaprojektowane w celu wymiany informacji graficznych między użytkownikami. Przyczyniły się one do rozwoju algorytmów modyfikujących bądź upiększających zdjęcia i materiały wideo [109, 54]. Ponadto wiele bibliotek zdecydowało się na wirtualizację swoich zasobów [50]. Ich zadanie jest o tyle wymagające, że wiele pozycji mogło zostać uszkodzonych wskutek użytkowania czy nieodpowiedniego ich przechowywania.

Systemy przetwarzające obrazy na tekst powinny cechować się dużą dokładnością, jak i uniwersalnością. Papier, kolor czy krój pisma mogą być zróżnicowane i istotnym jest, by takie systemy wspierały możliwie największą liczbę rozwiązań. Ponadto powinny one nie ograniczać się jedynie do alfabetu łacińskiego, lecz umożliwiać rozpoznawanie tekstów w różnych językach. Jest to istotne szczególnie w Polsce, gdzie język wykorzystuje znaki diakrytyczne niespotykane w językach bardziej popularnych. Brak wsparcia dla takich znaków może skutkować zmianą znaczenia niektórych wyrazów. Okazuje się, że w praktyce systemy automatycznej zamiany obrazów na tekst pozostawiają wiele do życzenia i wciąż nie nadają się do pełnej automatyzacji.

Niniejsza praca ma na celu przybliżyć metodę pozwalającą polepszyć jakość obrazów i jednocześnie ułatwić rozpoznawanie na nich tekstu. Przedstawia nową metodę usprawniania obrazu po binaryzacji. Zaprezentowany algorytm wykorzystuje morfologię sterowaną polem kierunkowym. Dzięki temu operacje morfologiczne dostosowują się do obszaru obrazu, który jest aktualnie przetwarzany. Proces doboru elementu strukturalnego w konkretnym regionie obrazu odbywa się na podstawie poziomu koherencji, z wykorzystaniem histogramu kołowego. Taki dobór kryterium pozwala zidentyfikować wiele linii w aktualnie przetwarzanej lokacji. Dobór kształtu oraz kierunku elementu pozwala zminimalizować liczbę opisanych powyżej proble-

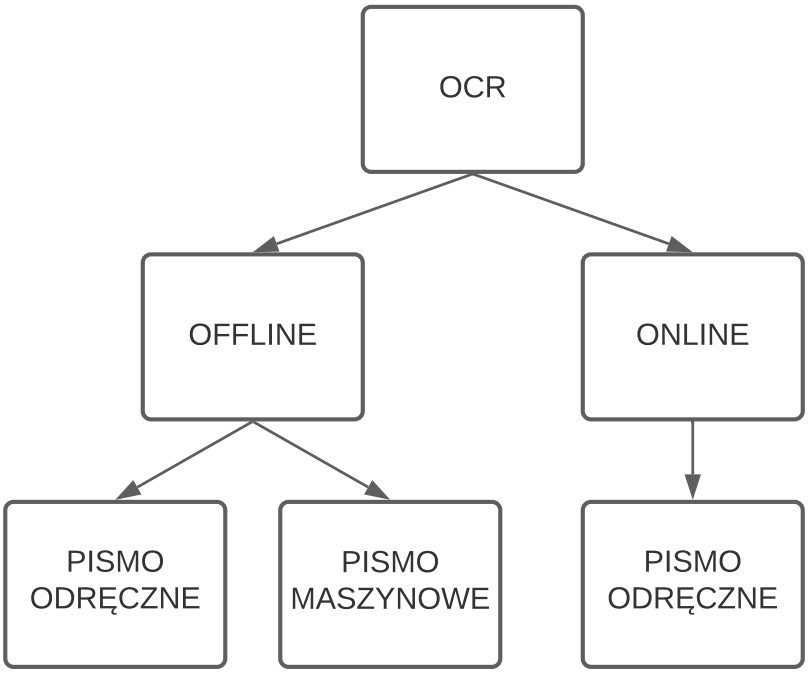
mów binaryzacji, takich jak wypełnianie pętli czy łączenie oddzielnych linii. Pomysłodawcą algorytmu był Marcin Adamski, wspólnie z autorem rozprawy i Khalidem Saeedem algorytm został opublikowany w 2021 r. [3]. Autor rozprawy zaimplementował algorytm i napisał cały kod, a także wykonał mnóstwo eksperymentów przeprowadzonych na różnych bazach obrazów, aby dostosować parametry do problemu rozpoznawania pisma i udowodnić skuteczność algorytmu w praktycznych zastosowaniach. Warto wspomnieć, że autor rozprawy opracował algorytmy, które z wysoką skutecznością rozpoznawały tekst na obrazach, jak na przykład w artykule z 2018 r. [83] czy z 2017 r. [105].

Przedstawiona metoda jest stosowana na obrazie zbinaryzowanym, przed zastosowaniem segmentacji. Literatura nie proponuje satysfakcjonującego określenia na ów etap w języku polskim, wobec czego w dalszej części pracy będzie on określany angielskim terminem jako preprocessing. Do poprawnego działania algorytm potrzebuje, poza obrazem zbinaryzowanym, także obrazu w odcieniach szarości. Jest on potrzebny do poprawnego znalezienia pola kierunkowego.

1 Wprowadzenie

1.1 Taksonomia systemów rozpoznawania pisma

Rozwój algorytmów przetwarzania obrazów z pismem ręcznym ma długą historię. Wiele przykładów kompletnych, w pełni funkcjonalnych systemów rozpoznawania pisma zostało opublikowanych w literaturze, a także wdrożonych do komercyjnych aplikacji. Metody rozpoznawania pisma (OCR) mogą zostać podzielone ze względu na naturę problemów, tzn. ze względu na typ analizowanych glifów oraz ze względu na ich akwizycję. Klasyczna taksonomia [25] została przedstawiona na Rysunku 1.



Rysunek 1: Taksonomia systemów rozpoznawania pisma

Kategoria online zawiera wszystkie algorytmy, które wykorzystują dodatkowe informacje, zebrane w czasie rzeczywistym podczas tworzenia napisu. Przykładem takich danych może być szybkość pisania, siła nacisku czy azymut. W celu zebrania takich informacji stosowane są specjalne narzędzia, takie jak elektroniczne tablety i inteligentne długopisy [78, 67].

Z drugiej strony kategoria offline skupia się jedynie na napisanym, statycznym tekście. Są to zazwyczaj zdjęcia czy skany dokumentów. W tym przypadku nie istnieje możliwość pobrania dodatkowych informacji (takich jak szybkość pisania), wobec tego jedyna możliwa metoda roz-

poznania polega na analizie kształtu glifów. W przypadku tekstu maszynowego, rozpoznawane są znaki poszczególnych czcionek komputerowych. W tym przypadku zazwyczaj poszczególne glify w tekście są łatwe do odseparowania. To sprawia, że algorytmy w tej kategorii osiągają skuteczności rozpoznawania wynoszące nawet do 99% [23].

Najtrudniejszą kategorią jest rozpoznawanie pisma odręcznego offline. W tym przypadku dodatkowym utrudnieniem jest częsty problem z odseparowaniem poszczególnych znaków, czy różne zdobienia pisma, uwzględniające charakterystyczne dla piszącego zaokrąglenia, pętelki. Ta kategoria jest często przedstawiana w literaturze jako ICR, aby odseparować ją od zacznie prostszego problemu rozpoznawania czcionek komputerowych. Szczególnym przypadkiem takich aplikacji są algorytmy rozpoznające podpis. Możemy wyszczególnić dwa cele tego typu algorytmów: weryfikacja osoby podpisującej się lub identyfikacja osoby, do której należy dany podpis [40]. Warto zaznaczyć, że w przypadku analizy pisma, zazwyczaj literatura porusza problem alfabetu łacińskiego. Kolejnym wyzwaniem jest więc uwzględnianie liter charakterystycznych dla danego języka, które często różnią się detalami od innego znaku alfabetu łacińskiego. W przypadku języka polskiego interesującym problemem jest znalezienie sposobu na rozróżnienie liter: *z*, *ż*, *ź* czy *o*, *ó*.

1.2 Charakterystyka algorytmów przetwarzania obrazów

W przypadku algorytmów przetwarzania obrazów można zidentyfikować pewien ogólny schemat funkcjonowania procesu. Został on przedstawiony na Rysunku 2.

Rysunek 2: Schemat algorytmów rozpoznawania obrazów

W przedstawionym schemacie można zidentyfikować następujące etapy: akwizycja danych, preprocessing, segmentacja, ekstrakcja cech oraz klasyfikacja.

W przypadku systemów działających offline, akwizycja danych ogranicza się jedynie do pobrania graficznej reprezentacji w postaci obrazu rastrowego. Jest to etap, w którym krystalizuje się poziom trudności problemu. Skomplikowanie zagadnienia zależy od rozdzielczości obrazu, czy wykorzystywanej przestrzeni barw. W szczególności dla problemów medycznych

przestrzeń barw jest często mocno ograniczona, co znacząco ogranicza liczbę algorytmów, które można zastosować na tego typu obrazach [61].

W przypadku operowania na istniejącej bazie danych, preprocessing jest pierwszym etapem, w którym następuje automatyzacja przetwarzania obrazu. Od jakości tego etapu zależy jakość segmentacji, ekstrakcji cech i w konsekwencji klasyfikacji. W przypadku zastosowania nieodpowiedniej metody preprocessingu, możliwa jest utrata części informacji, przez co wyekstrahowanie cech może okazać się niemożliwe. Jednocześnie jest to etap trudno mierzalny, w którym często stosuje się prymitywne metody. Zazwyczaj nie dają one dobrych rezultatów, co zostanie wykazane w niniejszej pracy. Najważniejszą operacją w etapie preprocessingu jest binaryzacja. Dzieli ona wszystkie piksele obrazu na dwie klasy: reprezentującą tło albo pismo. Jest to operacja kluczowa dla pozostałych etapów, mająca duży wpływ na wyniki całego algorytmu. Niestety, podczas tego etapu obraz traci informacje, które mogą być kluczowe do poprawnego rozpoznania poszczególnych znaków. Do najczęstszych niepożądanych skutków należą zdeformowane linie oraz uwidocznienie szumu i zaklasyfikowanie go jako część pisma. Tego typu artefakty uniemożliwiają poprawne działanie algorytmów używających obrazu po binaryzacji jako wejściowego. Przykładami algorytmów czułych na takie błędy są: cieniowanie, szkieletyzacja bądź operacja śledzenia linii. Mimo wielu lat badań trwających nad algorytmami binaryzacji, problem zdaje się wciąż nie mieć satysfakcjonującego rozwiązania. Dostępne metody zdają się cierpieć na różne problemy, które są dodatkowo potęgowane w przypadku przetwarzania zdjęć o niskiej rozdzielczości [17, 75]. Z tego powodu, aby zniwelować negatywne skutki binaryzacji, często stosuje się dodatkowe algorytmy na obraz po binaryzacji. Przykładem takich operacji jest filtr medianowy oraz operacje morfologiczne bazujące na dylatacji czy erozji. Jednak problemem przy stosowaniu tych filtrów jest generowanie nowych błędów, które w szczególnych przypadkach mogą nawet pogorszyć obszar obrazu poprawnie przetworzony przez algorytm binaryzacji. W przypadku obrazów podpisów, szczególnie dotkliwym problemem jest łączenie równoległych linii, wypełnianie celowo utworzonych pętli czy przerwanie ciągłości linii.

Proces segmentacji, w przypadku analizy tekstu, polega na wydzieleniu poszczególnych glifów składających się na słowo. W przypadku tekstu maszynowego poszczególne znaki są zazwyczaj od siebie oddzielone, co czyni ten etap bardzo prostym. Przy założeniu poprawnego preprocessingu, proces ogranicza się do zidentyfikowania spójnych elementów, nie będących tłem. Problem jest dużo bardziej zaawansowany w przypadku tekstu odręcznego. W tym przy-

padku należy poprawnie zidentyfikować łączenia liter, a następnie podzielić jeden spójny fragment na podproblemy - rozpoznawanie poszczególnych znaków. Z kolei przetwarzając podpisy, często niemożliwym jest wyodrębnienie poszczególnych glifów - wówczas należy zidentyfikować elementy charakterystyczne dla podpisu.

Ekstrakcja cech określa, w jaki sposób glif będzie reprezentowany podczas klasyfikacji. Literatura proponuje różne podejścia do tego zagadnienia [85, 22, 55, 41]. W przypadku pisma maszynowego, najlepsze algorytmy są niemal bezbłędne. Szczególnie skuteczne są w tym przypadku sieci neuronowe, które mogą zostać nauczone na zbiorach obrazów czcionek komputerowych [89]. Jednak takie podejście zakłada konieczność wydzielenia nie tylko poszczególnych linii w tekście, lecz także odseparowania od siebie kolejnych glifów. Czcionki maszynowe charakteryzują się zazwyczaj prostą strukturą i taka operacja nie jest wymagająca. Jednak istnieją czcionki ozdobne, oferujące łączenia liter. Takie nieprawidłowe odseparowanie skutkuje przypisaniem nieprawidłowych cech do poszczególnych klas liter. Próba manualnego naprawienia tak nauczonej sieci po procesie segmentacji zazwyczaj nie jest możliwa [48].

Warto zauważyć, że w zależności od problemu, algorytm może kończyć pracę już na etapie preprocessingu. Taka sytuacja ma miejsce, gdy powstały obraz ma zostać przeanalizowany przez specjalistę z wiedzą a priori. Wówczas celem algorytmu jest takie przetworzenie obrazu, które wyodrębni elementy szczególnie istotne dla specjalisty i tym samym ułatwi mu dalszą pracę.

1.3 Cel i teza rozprawy

Celem niniejszej pracy jest udowodnienie tezy, że istnieje uniwersalne podejście do preprocessnigu obrazów zawierających obiekty o liniowej strukturze, które poprawiałoby wyniki zwracane przez binaryzację. Metoda powinna polepszać wyniki dowolnego algorytmu binaryzacji i naprawiać błędy typowe dla struktur liniowych, takie jak przerwane czy zdeformowane linie.

Ocena skuteczności powinna bazować zarówno na analizie wizualnej, jak i na analitycznej mierze, bazującej na obrazach wzorcowych.

1.4 Struktura pracy

Rozdział 2 zawiera opis aktualnie dostępnej literatury z zakresu rozpoznawania pisma i obiektów o strukturze liniowej. Przedstawiono algorytmy z zakresu preprocessingu, które proponują różne podejścia do rozwiązania problemu. Dzięki temu można zaobserwować sposób, w jaki

zmieniały się owe algorytmy, ich mocne i słabe strony. Można także wyznaczyć pewne klasyczne schematy podejścia do preprocessingu, które warto mieć na uwadze podczas tworzenia nowego rozwiązania.

W rozdziale 3 można znaleźć opis nowego, autorskiego podejścia do preprocessingu. Dzięki swojej modularności, w prezentowanym rozwiązaniu można zidentyfikować komponenty wzorujące się na istniejących rozwiązaniach opisanych w poprzednim rozdziale bądź rozszerzające je.

Rozdział 3.1 prezentuje schemat całej metody, oczekiwane wejście, przewidywane wyjście. Opisuje etapy istniejące w algorytmie oraz prezentuje kilka sposobów pozwalających zaimplementować każdy z etapów.

W rozdziale 3.2 przedstawiono algorytmy binaryzacji najczęściej wykorzystywane w przeanalizowanej literaturze przy strukturach liniowych. Każda z metod ma swoje wady oraz zalety, które zostały zidentyfikowane w tym rozdziale. Każdy ze znalezionych problemów powinien znaleźć swoje rozwiązanie w autorskiej metodzie.

Rozdział 3.3 opisuje operacje morfologiczne często stosowane w literaturze. Stosowanie pojedynczej metody daje zazwyczaj niezadowalające rezultaty. Jednak ich połączenie i odpowiednie wykorzystanie pozwala polepszyć obraz wejściowy, czego dowodem jest autorski algorytm.

Istotnym elementem algorytmu jest wyznaczenie pola kierunkowego. Autorskie rozwiązanie nie narzuca konkretnej metody, stąd rozdział 3.4 proponuje kilka opcji na wyznaczenie pola kierunkowego. W dalszej części niniejszej rozprawy każda z tych metod zostanie przeanalizowana pod kątem skuteczności działania.

Test 4-sąsiedztwa, opisany w rozdziale 3.5, opisuje opcjonalny komponent algorytmu, pozwalający w prosty sposób pozbyć się nadmiaru szumu. Literatura pokazuje, że w niektórych problemach daje rewolucyjne rezultaty i może być stosowany jako jedyna metoda preprocessingu.

Kolejnym opcjonalnym modułem autorskiego algorytmu jest zwiększenie liczby próbek. Algorytm został opisany w rozdziale 3.6. Może on być przydatny szczególnie w przypadku obrazów o niskiej rozdzielczości. Metoda wspomaga działanie innych algorytmów np. pozawala znaleźć więcej możliwości podczas wyznaczania pola kierunkowego.

Rozdział 4 skupia się na opisie sposobów weryfikacji prezentowanego algorytmu. Z racji jego modularności, wymaga to dokładnego zaplanowania sposobu testowania, aby uzyskać

rzetelne wyniki. W tym celu zostały zaproponowane dwie miary dokładności. Każda z nich została opisana oraz przeanalizowana pod kątem praktyczności uzyskanych rezultatów dla obrazów przedstawiających pismo odręczne.

Następny rozdział 5 zawiera trzy rodzaje testów: wizualnych, analitycznych i statystycznych. W celu umożliwienia odtworzenia uzyskanych rezultatów, przedstawiono listę parametrów użytych podczas testów na wybranych bazach obrazów. W rozdziale zostały także uwzględnione porównania autorskiego rozwiązania z innymi algorytmami używanymi w praktyce. Zostało porównane rozwiązanie oparte na sieciach konwolucyjnych, a także filtr Frangi'ego - jeden z popularniejszych algorytmów wykorzystywanych podczas przetwarzania obrazów medycznych.

Rozdział 6 przedstawia wnioski wynikające z przedstawionych testów. W tym miejscu zostały także opisane praktyczne sytuacje, w których autorski algorytm może być wykorzystywany. Opisane zostały także potencjalne możliwości rozszerzenia algorytmu.

2 Powiązane prace i stan wiedzy

Preprocessing obrazu jest ciągiem metod, który ma na celu poprawić jakość obrazu i uwydatnić jego cechy, potrzebne podczas dalszego procesowania. W szczególności takie operacje doprowadzają do rozpoznania obiektu. Etap preprocessingu ma na celu wyeliminowanie szumów powstałych wskutek akwizycji i ułatwienie identyfikacji cech kluczowych przy dalszych etapach. Literatura pokazuje wiele podejść do preprocessingu obrazu. W [8] został przedstawiony zbiór metod, które można podzielić na następujące kategorie:

- metody prebinaryzacyjne,
- binaryzacja,
- metody postbinaryzacyjne.

Pierwsza kategoria zawiera algorytmy, które operują na obrazie w odcieniach szarości. Do zbioru tych metod należy: wyrównanie histogramu, normalizacja histogramu, dostosowanie jasności lub kontrastu na obrazie. Każda z tych operacji wymaga wiedzy statystycznej na temat jasności pikseli czy liczby pikseli o danej jasności. Jednak w przypadku tych metod samo rozmieszczenie pikseli nie ma znaczenia, ta funkcja jest globalnie wykonywana dla każdego piksela na obrazie, niezależnie od jego sąsiadów.

Binaryzacja jest kluczowym etapem nie tylko dla preprocessingu, ale dla całego procesu przetwarzania obrazu. To na tym etapie dokonuje się podział całego zbioru pikseli na tło oraz istotne informacje z punktu widzenia dalszego przetwarzania. Podczas tego etapu dokonuje się znaczne uproszczenie obrazu wejściowego. Wiąże się to z redukcją informacji na samym obrazie. Poprawnie dokonany proces binaryzacji powinien zachować wszystkie kluczowe informacje oraz skutecznie odseparować piksele tła od pozostałych - pikseli obiektu. Jest to nietrywialne zadanie, w praktyce algorytm binaryzacji dostosowuje się do aktualnego zadania. Mimo to w większości zastosowań sama binaryzacja nie daje zadowalających efektów [13].

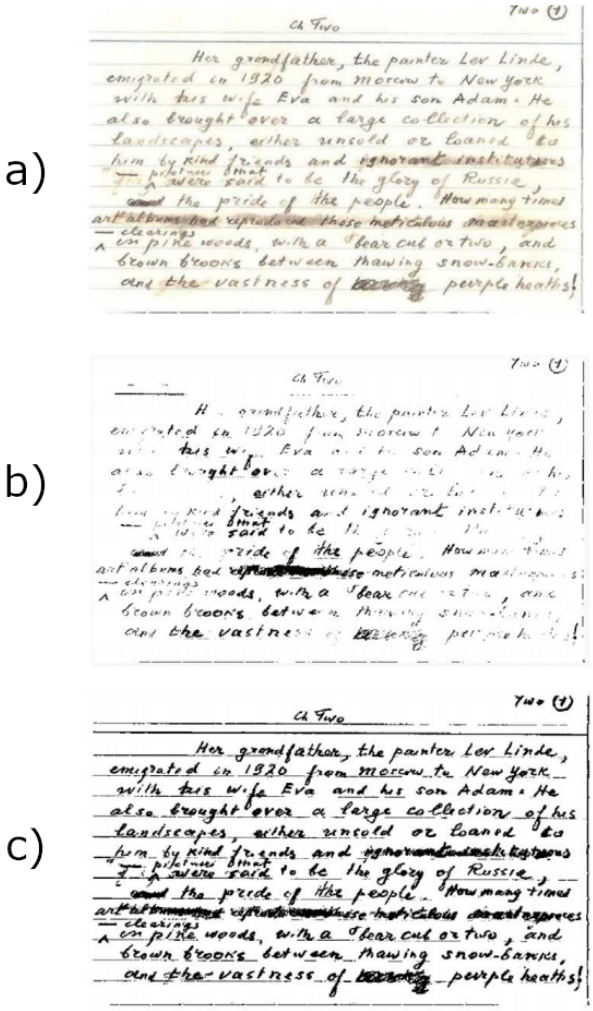
Aby zniwelować błędy binaryzacji, stosowane są algorytmy postbinaryzacyjne, mające na celu naprawić błędy powstałe wskutek binaryzacji. Należą do nich operacje morfologiczne. Są to algorytmy bazujące na teorii mnogości, mające na celu zmodyfikować proporcje liczby pikseli obiektu do liczby pikseli tła [91]. W przeciwieństwie do metod prebinaryzacyjnych, te zazwyczaj charakteryzują się działaniem lokalnym. Efekt ich oddziaływania na dany piksel jest zależny od sąsiedztwa, w jakim ten piksel się znajduje. Najczęściej spotykanymi operacjami są

dylatacja i erozja, a także różne ich modyfikacje. Szersze omówienie tych metod znajduje się w rozdziale 3.3. Inną metodą preprocessingu jest filtr medianowy. Ma on szerokie zastosowanie z racji relatywnie dobrych efektów w porównaniu do szybkości działania. Poza modyfikacją pola obiektu na obrazie, często spotykanymi operacjami są metody gradientowe. Umożliwiają one identyfikację krawędzi na obrazie [92]. Jest to operacja szczególnie praktyczna podczas segmentacji wielu obiektów z obrazu.

Należy zaznaczyć, że czasem literatura nazywa cały proces preprocessingu binaryzacją [106, 104]. Niniejsza rozprawa doktorska stawia sobie za cel poprawianie jakości obrazu bez względu na wybraną metodę binaryzacji. Stąd sam proces binaryzacji będzie określany jako nadanie każdemu pikselowi obrazu jednej z dwóch możliwych wartości. Etapy prebinaryzacji i postbinaryzacji nie zawsze są obecne, jednak zazwyczaj ich brak rzutuje na jakość algorytmu, co udowadniają wyniki klasyfikacji.

Literatura pokazuje, że istnieje wiele podejść do preprocessingu tekstu pisanego ręcznie [62]. Przykładowo, w konkursie DIBCO (Competition on Document Image Binarisation) większość przesłanych algorytmów uwzględnia w swoim działaniu poprawianie jakości obrazu. W szczególności, filtry są stosowane zarówno przed binaryzacją, jak i po niej [75]. Jednak problemem, jaki da się zauważyć przy stosowaniu tych algorytmów jest fakt, że współpracują one jedynie z konkretnym typem binaryzacji. Autorzy tych rozwiązań często nie podają szczegółowych informacji na temat parametryzacji ich metod, co znacząco utrudnia dostosowanie ich do konkretnego przypadku.

Ciekawym przypadkiem może być algorytm zaprezentowany w [18]. Proponuje on kombinację trzech technik, aby poprawić efekty binaryzacji. Są to: dyfuzja anizotropowa do redukcji plam-szumów, wypełnianie luk, bazując na punktach charakterystycznych tekstu oraz uwydatnienie tekstu poprzez filtry rozszerzające. Efekty takiego podejścia można zobaczyć na Rysunku 3. Można na nim dostrzec problemy wynikające ze stosowania jedynie binaryzacji. Nierówne oświetlenie powoduje powstanie niedokładności podczas binaryzacji poszczególnych liter. Informacja na temat części z nich, znajdujących się w prawym górnym rogu na Rysunku 3 b), została utracona. Taki błąd można wyeliminować poprzez dostosowanie metody binaryzacji do problemu. Przykładowo dużo lepsze efekty w tym przypadku Autorzy uzyskali poprzez zastosowanie binaryzacji będącej modyfikacją adaptacyjnego progowania Wellnera [18]. Innym problemem widocznym na Rysunku 3 są pojedyncze piksele błędnie zaklasyfikowane jako piksele nie należące do tła. Przykładowo, część linii kartki bądź ciemniejszy kolor kartki w jednej z



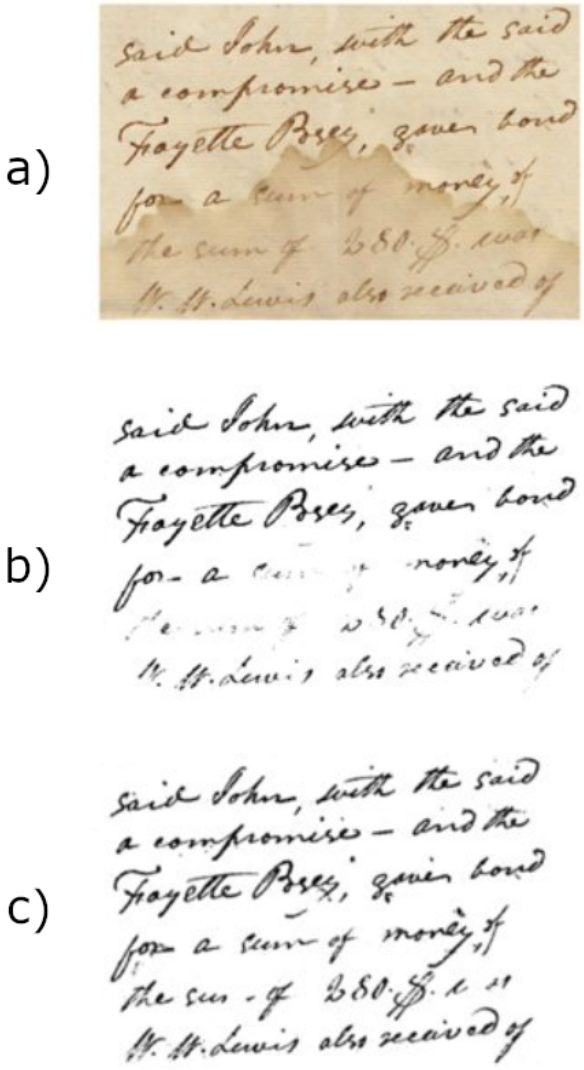
Rysunek 3: a) Oryginalny obraz
b) Obraz po binaryzacji Sauvoli
c) Obraz algorytmu opartego o dyfuzję anizotropową, wypełnianie luk i filtry rozszerzające
Źródło: [18]

jej środkowych linii powoduje błędy binaryzacji. W celu eliminacji tych błędów użyto dyfuzji anizotropowej, która skutecznie wyeliminowała pojedyncze artefakty. Dalsze badania Autorów sugerują, że takie podejście wpływa negatywnie na ciągłość linii. Te z kolei są naprawiane poprzez analizę szkieletu obiektów. Jest to rozwiązanie podobne do zastosowanego w [83].

Podobna technika została wykorzystana w [36]. Autorzy identyfikowali obszary 8-spójne i jeśli liczba pikseli w takim obszarze była mniejsza od ustalonego progu, taki obszar był usuwany. Jest to ciekawa alternatywa dla stosowania dyfuzji anizotropowej. Cel obu metod jest taki sam, natomiast metoda analizy obszarów 8-spójnych jest mniej kosztowna wydajnościowo.

Z kolei w [86], zostały użyte operacje zamknięcia i otwarcia. W tym przypadku obrazy

zostały zbinaryzowane metodą bazującą na cechach wyekstrahowanych z obrazu przy pomocy banku filtrów Gabora. Rezultaty zostały przedstawione na Rysunku 4. W celu uzyskania prze-

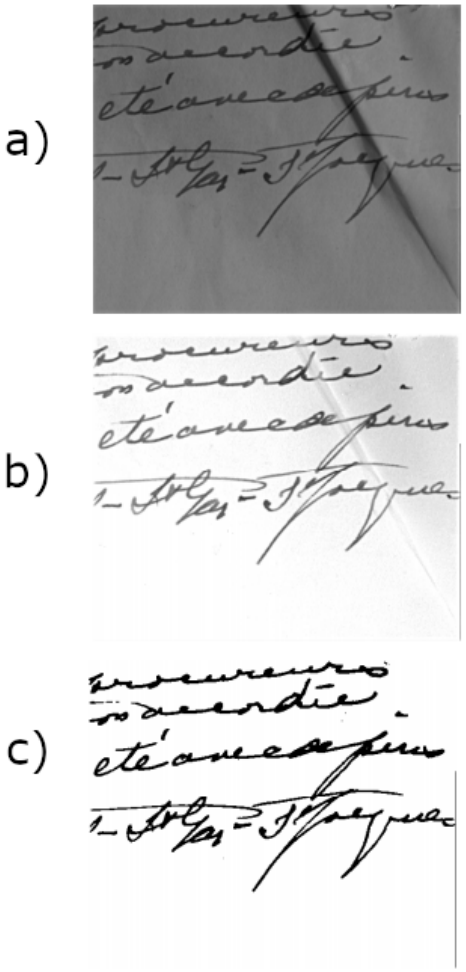


Rysunek 4: a) Oryginalny obraz
b) Obraz po binaryzacji Sauvoli
c) Obraz algorytmu opartego o bank filtrów Gabora
Źródło: [86]

wagi zwykłym algorytmem postbinaryzacyjnym, Autorzy wykorzystują w swoim podejściu informację na temat kierunku obiektu na obrazie w odcieniach szarości. Autorzy zauważyli, że gradient obliczany przez filtr Gabora zachowuje się inaczej dla pikseli obiektu, a inaczej dla pikseli tła. Ta obserwacja umożliwiła przetwarzanie obrazu po binaryzacji. Piksele, które zostały zaklasyfikowane jako tło zostały poprawione na piksele obiektu, jeśli informacja zebrana z obrazu w odcieniach szarości o tym świadczyła. W analogiczny sposób niepoprawne piksele

obiektu są zamieniane na piksele tła.

Z kolei w [63], Autorzy wykorzystali dodatkową wiedzę na temat obrazu. Poza standardowymi kanałami RGB, Autorzy wzięli pod uwagę szersze długości fali światła, otrzymując obraz w reprezentacji podczerwieni, a także ultrafioletu. By wyeliminować artefakty powstałe w wyniku binaryzacji, bazowali na operacjach sumy i różnicy logicznej, osiągając rezultat widoczny na Rysunku 5. W tym przypadku algorytm skupia swój punkt ciężkości na prebinaryzacji. Jest



Rysunek 5: a) Oryginalny obraz
b) Obraz po wyekstrahowaniu pikseli obiektu
c) Obraz po zastosowaniu binaryzacji
Źródło: [63]

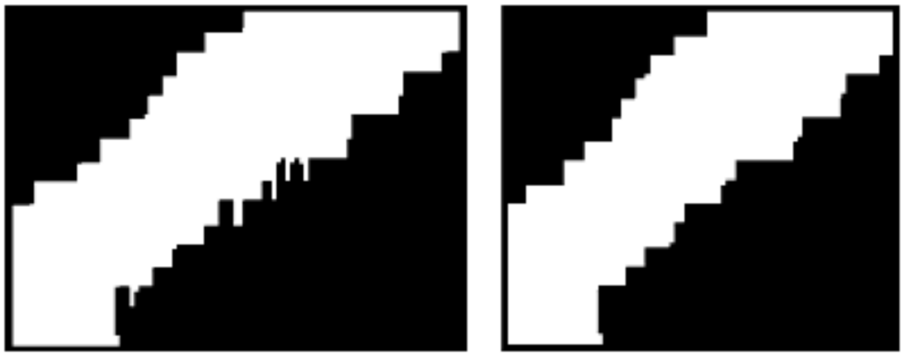
to możliwe dzięki dostępowi do większej ilości informacji w porównaniu ze zwykłym zdjęciem mającym trzy kanały RGB lub jeden reprezentujący odcienie szarości. To pozwoliło Autorom stworzyć histogram, na podstawie którego uzyskują precyzyjną informację o pikselach tła i obiektu. Dzięki temu binaryzacja następuje nie na podstawie kanału odcieni szarości, lecz na

podstawie długości fali światła. W taki sposób Autorzy są w stanie uzyskać piksele tła, które są następnie odejmowane od pierwotnego obrazu. Rezultatem takiej operacji są piksele reprezentujące pismo. Takie rozwiązanie daje bardzo obiecujące efekty i w przypadku dostępu do tak rozszerzonej informacji na temat obrazu warto zastosować ten algorytm. Niestety, w aktualnie sprzedawanych aparatach fotograficznych nie jest to popularna funkcjonalność.

Z kolei w [37] zastosowano nowatorskie podejście z wykorzystaniem sieci konwolucyjnych. W rezultacie metoda zwróciła znacznie poprawiony efekt binaryzacji Otsu. Jednak podany algorytm działa poprawnie jedynie z tym konkretnym typem binaryzacji. Co więcej, podobne rezultaty mogą zostać osiągnięte z wykorzystaniem bardziej uniwersalnych algorytmów, jak na przykład ten zaprezentowany w niniejszej rozprawie.

Mimo że jest wiele badań na temat różnych podejść do ekstrakcji cech z podpisów oraz klasyfikacji, niewiele badań skupia się na problemie preprocessingu tych obrazów [24, 40]. W celu pozbycia się błędów binaryzacji, większość prac stosuje prosty filtr medianowy [10, 46], filtr uśredniający [39] czy podstawowe operacje morfologiczne, takie jak: dylatacja, erozja oraz ich superpozycje, takie jak otwarcie i zamknięcie [30, 39].

Inny rodzaj preprocessingu zakłada usuwanie „postrzępionych" krawędzi powstałych wskutek binaryzacji. W [29] takie artefakty zostały wyeliminowane poprzez konwertowanie pikseli podpisu na piksele tła w sytuacji, gdy nie był spełniony warunek, że górny i dolny lub lewy i prawy piksel to piksele pisma. Przykładowy efekt takiego działania przedstawia Rysunek 6. Zmiany, jakich dokonuje ten algorytm są niewielkie. Wynika to z faktu analizy niewielkiego



Rysunek 6: Efekt działania algorytmu wygładzającego postrzępione krawędzie. Po lewej: obraz przed zastosowaniem algorytmu, po prawej: obraz bez postrzępionej krawędzi
Źródło: [39]

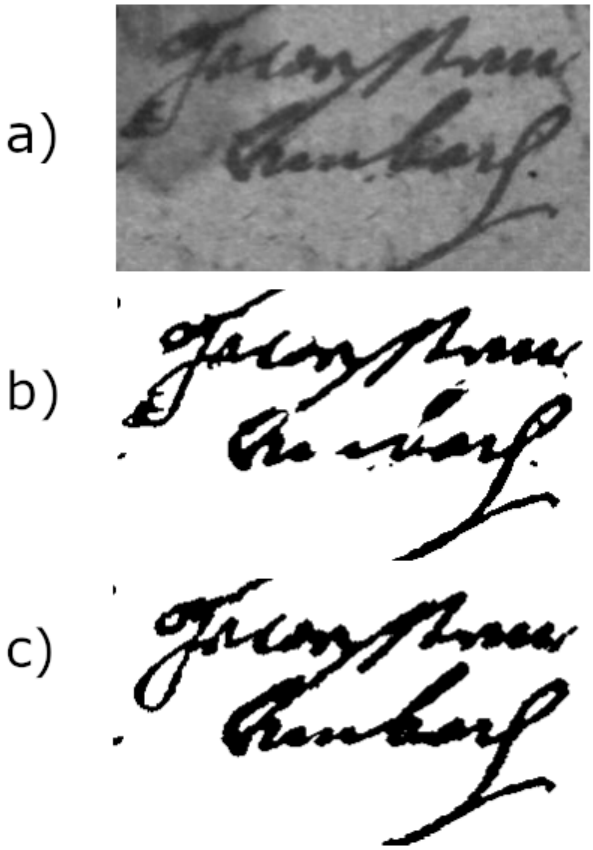
regionu - jedynie pikseli najbliższych aktualnie przetwarzanemu. Skuteczność działania takiego podejścia zależy także od rozdzielczości obrazu wejściowego. Należałoby przeanalizować, w

jaki sposób zwiększenie analizowanego obszaru wpływa na ostateczny rezultat. Na bazie przeprowadzonych doświadczeń można wysnuć wniosek, że algorytm nieznacznie poprawia jakość obrazu poprzez zwiększenie jej klasy ciągłości. Ma to zastosowanie w szczególności w algorytmach identyfikujących krawędzie. Należy jednak mieć na uwadze, że ta metoda ma największe zastosowanie, gdy jest wykorzystana razem z innymi, skuteczniejszymi podejściami.

Inne warte uwagi podejście do filtrowania obrazów zostało zaprezentowane w [102]. Użyto w nim zróżnicowanych elementów strukturalnych, które zostały użyte do obrazu w odcieniach szarości oraz po binaryzacji. Jednak autorzy [102] wykorzystali inne podejście przy projektowaniu pola kierunkowego oraz wyborze na tej podstawie elementu strukturalnego. W przytoczonej pracy pole kierunkowe zostało oszacowane przy użyciu średniego gradientu a następnie zastosowano element strukturalny zgodny kierunkiem wektora [3]. Metoda umożliwia obliczenie kierunku w każdymi punkcie obrazu poprzez propagację wartości gradientu z krawędzi, gdzie jest on dobrze określony. Warto zwrócić uwagę na zastosowany element strukturalny. Ma on kształt prostokąta, który zmienia długości swoich boków w zależności od dystansu od krawędzi: od elementu liniowego (tuż przy krawędzi) do kwadratu (przy większym dystansie). Jednak takie podejście nie daje zadowalających rezultatów, gdy jest zastosowane do zdjęć zawierających pismo odręczne. Z racji faktu, że litery w tekście zazwyczaj mają małą szerokość, propagacja gradientu zwykle nie jest potrzebna, co więcej, może powodować problemy w przypadku przetwarzania obrazów o niskiej rozdzielczości i dużej liczbie szumów. Poprawność działania algorytmu zależna jest od poprawnego zidentyfikowania krawędzi – jeśli na tym etapie pojawi się błąd, może to skutkować wyborem nieprawidłowego elementu strukturalnego, co przełoży się na zdeformowanie kształtu poszczególnych znaków, zamiast ich naprawienia. Ponadto wyniki przedstawione przez autorów [102] nie zawierają porównania z obrazami wzorcowymi, a zawierają jedynie wizualną ocenę. To utrudnia obiektywną ocenę algorytmu.

Kolejne podejście opiera się na wykorzystaniu sieci konwolucyjnych. Takie algorytmy zazwyczaj nie rozróżniają poszczególnych etapów w preprocessingu, są więc mniej elastyczne [98, 4, 70]. Oferują jednocześnie binaryzację oraz poprawianie jakości obrazu. Zaletą takiego podejścia jest wysoka skuteczność rozwiązywania problemu, na którym sieci były trenowane. Jednak takie rozwiązania nie sprawdzają się jako uniwersalne algorytmy poprawiające jakość obrazu. W szczególności algorytmy nauczane na alfabecie łacińskim nie gwarantują wysokiej skuteczności w przypadku innych alfabetów. W [98] zaprezentowano algorytm nauczony na bazie DIBCO [1]. Wykorzystano w nim funkcję określającą prawdopodobieństwo każdego piksela

na bycie pikselem tła. Autorzy zauważyli, że podczas standardowej procedury dzielenia obrazu na coraz mniejsze fragmenty, algorytm znacznie lepiej sobie radzi, jeśli nastąpi zwiększenie częstości próbkowania dla tych mniejszych obrazów. Takie podejście pozwala uzyskać skuteczność ok. 90% na trudnej do procesowania bazie. Przykładowy efekt został przedstawiony na Rysunku 7. Mimo nieregularnego tła, algorytm dobrze poradził sobie z przetworzeniem przy-

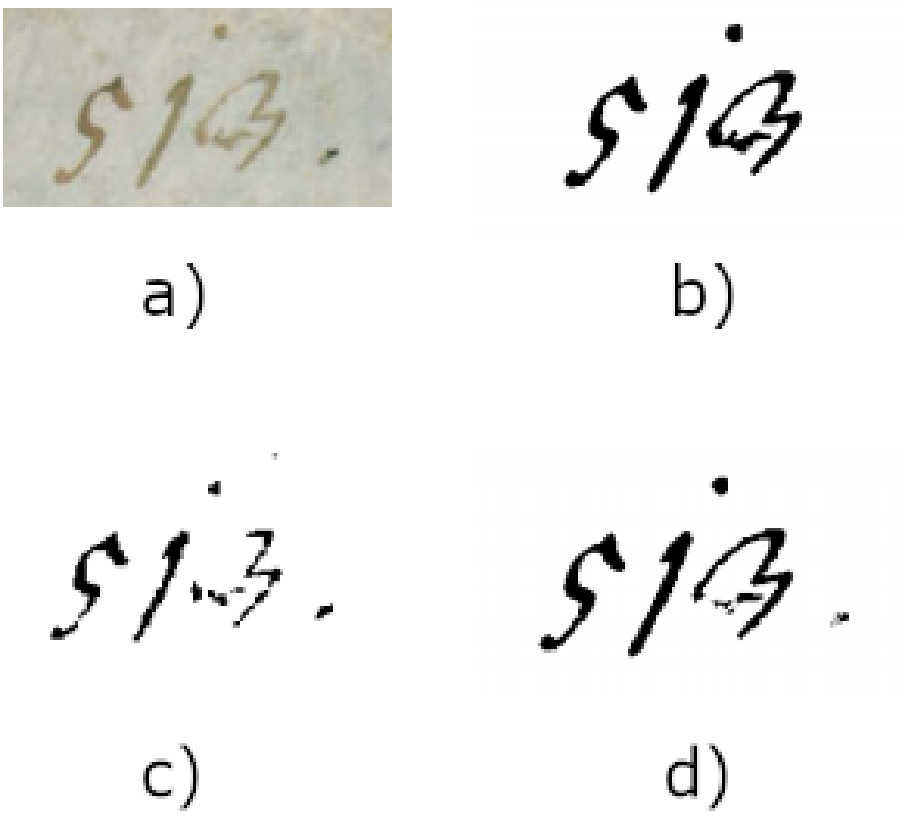


Rysunek 7: a) Obraz w odcieniach szarości b) Obraz przetworzony przez algorytm sieci konwolucyjnych ze zwiększeniem próbkowania
c) Oczekiwany rezultat preprocessingu
Źródło: [98]

kładowego zdjęcia. Warto zwrócić uwagę na pojedyncze artefakty widoczne w środkowej części obrazu. Mogłyby one zostać wyeliminowane, jeśli po tym algorytmie byłaby użyta metoda postbinaryzacyjna. Pierwsza litera w drugim rzędzie została w niepoprawny sposób przetworzona. Oczekiwanym rezultatem było utworzenie dwóch równolegle biegnących linii. Natomiast algorytm sieci konwolucyjnych połączył obie linie, tworząc nieoczekiwaną pętlę. Jednak mimo drobnych błędów, Autorzy chwalą się wysoką skutecznością. Zauważają także, że do nauki, sieć potrzebuje dużej liczby obrazów testowych. Warto zauważyć, że mniejszą skuteczność ma

mała liczba obrazów o dużej rozdzielczości - algorytm potrzebuje zróżnicowanych przypadków testowych do nauki. To znacząco ogranicza praktyczność zastosowania algorytmu w wielu dziedzinach nauki i jest główną przyczyną nie używania powszechniej algorytmów bazujących na sieciach konwolucyjnych.

Inny przykład algorytmu wykorzystującego sieci konwolucyjne został przedstawiony w [4]. Autorzy wykorzystali kolorowe obrazy z bazy DIBCO [1]. Zaprojektowali wiele niezależnych od siebie sieci, każda odpowiadająca za osobny kanał koloru. Zdjęcie będące rezultatem przetwarzania takiego algorytmu jest utworzone poprzez zastosowanie operacji minimum na obrazach wyjściowych każdej z sieci. Efekt działania takiego algorytmu na przykładowym zdjęciu został przedstawiony na Rysunku 8. Daje on wyniki zbliżone do algorytmu wzorcowego. Zde-



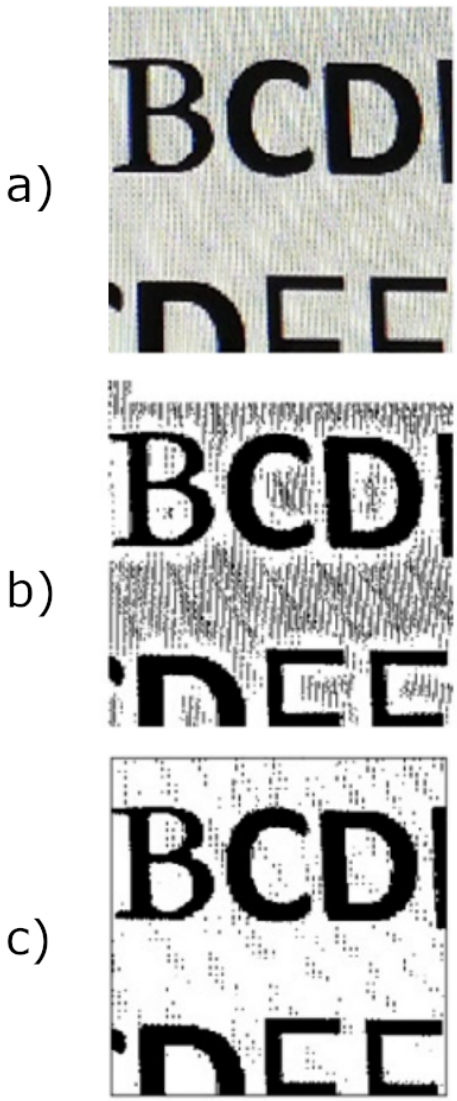
Rysunek 8: a) Obraz w odcieniach szarości b) Oczekiwany rezultat preprocessingu
c) Obraz po binaryzacji metodą Otsu
d) Obraz przetworzony przez algorytm z siecią konwolucyjną przypisaną do każdego kanału koloru Źródło: [4]

cydowanie oferuje lepszy wynik w porównaniu do binaryzacji bez stosowania technik postbinaryzacji. Co ciekawe, binaryzacja Otsu zazwyczaj charakteryzuje się pogrubieniem obiektu.

W tym przypadku grubsze linie pisma zostały uzyskane przez algorytm wykorzystujący sieci konwolucyjne. Jest to związane ze zbliżonym kolorem tuszu oraz tła. Do metody Otsu takie zdjęcie zostało przekonwertowane na obraz w odcieniach szarości, posiadający jeden kanał, informujący o intensywności piksela. Algorytm [4] jest dostosowany do wykorzystania większej ilości informacji, więc przeanalizował wszystkie kanały. Takie podejście sprawdza się jedynie w bazach danych, które posiadają kolorowe zdjęcia. Dla obrazów w odcieniach szarości ów algorytm traci swoją przewagę i staje się klasycznym podejściem do implementacji sieci konwolucyjnych.

Ciekawą modyfikacją zagadnienia analizy obrazów pisma jest uwzględnienie efektu moiré. Jest to efekt powstawania prążków wskutek interferencji dwóch zbiorów linii. Jest on możliwy do zaobserwowania podczas robienia zdjęcia ekranu urządzenia elektronicznego. Przykład takiego efektu został przedstawiony na Rysunku 9. Można na nim zauważyć, że zastosowanie binaryzacji na takim obrazie generuje dużą ilość szumów. Wobec tego należy wykorzystać metodę prebinaryzacji, by wyeliminować problem jeszcze przed procesem binaryzacji. Jednym z możliwych rozwiązań tego problemu jest zastosowanie filtru Gaussa. Jest to filtr dolnoprzepustowy, który powoduje rozmazanie obrazu. Teoria mówi, że potrafi on poprawić jakość obrazu przy dużej ilości szumów [32]. Jednak zależne jest to od współczynnika SNR - stosunku intensywności sygnału do szumu. W przypadku Rysunku 9 zastosowanie filtru Gaussa pozwala utworzyć jednorodną powierzchnię tła. Jednak wiąże się to z rozmazaniem krawędzi pisma. Taki sposób spowoduje więc dodatkowe problemy, wobec tego nie powinien być stosowany. Inne rozwiązanie zostało zaproponowane w [35]. Autorzy użyli sieci konwolucyjnej, która dokonuje równoległych obliczeń na pięciu różnych rozdzielczościach obrazu. Poprzez porównanie rezultatów możliwe jest wyeliminowanie zaistniałego efektu bez pogarszania stanu obiektów znajdujących się na obrazie.

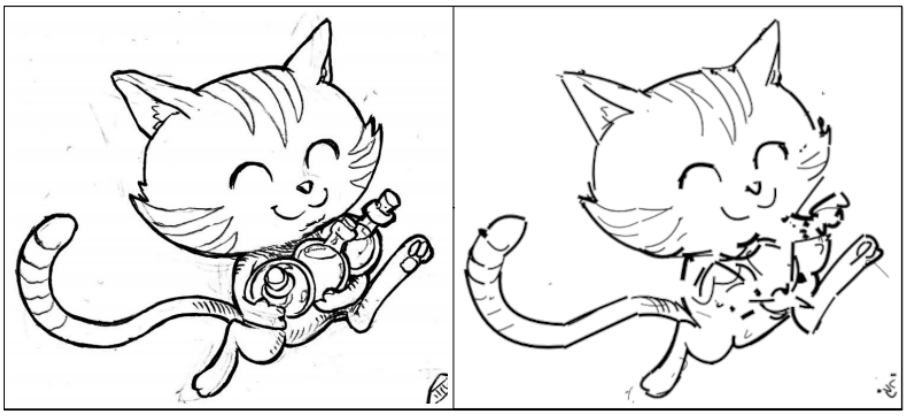
Ekstrakcja liniowych struktur jest ważnym zagadnieniem w problemie wektoryzacji obrazów rastrowych. Pozornie może się wydawać że problem wektoryzacji jest osobnym zagadnieniem w porównaniu do problemu omawianego w niniejszej pracy. Algorytmy wektoryzacji mają na celu wygenerować z obrazu możliwie jak najmniej skomplikowaną reprezentację kształtów, opisaną przy pomocy wektorów czy krzywych Beziera [26, 27]. Mimo to, takie algorytmy również spotykają się z problemem segmentacji pożądanych kształtów od tła. Owa segmentacja jest prowadzona jako początkowy etap wektoryzacji i jakość tego procesu znacząco wpływa na jakość samej wektoryzacji. Przykładowo, w [27] cała segmentacja została wykonana za po-



Rysunek 9: a) Oryginalny obraz z efektem moiré
b) Obraz po binaryzacji metodą Sauvoli
c) Obraz po binaryzacji metodą Otsu
Źródło: [35]

mocą sieci konwolucyjnej U-net, która przyporządkowuje każdy piksel do jednej z dwóch klas, była trenowana na zdjęciach oraz grafikach. Jest to technika zbliżona do binaryzacji z użyciem sieci konwolucyjnych [37, 104]. Takie metody mogą dawać zadowalające wyniki, gdy są trenowane na parach obrazów: w odcieniach szarości oraz oczekiwany rezultat binaryzacji. Jednak zbiór treningowy jest głównym problemem w przypadku sieci CNN. Powinien być odpowiednio duży i reprezentatywny. W praktyce użyteczność metod opartych na uczeniu maszynowym w tym przypadku jest ograniczona. Przykładowo, jeśli sieć była uczona na obrazach rysunków

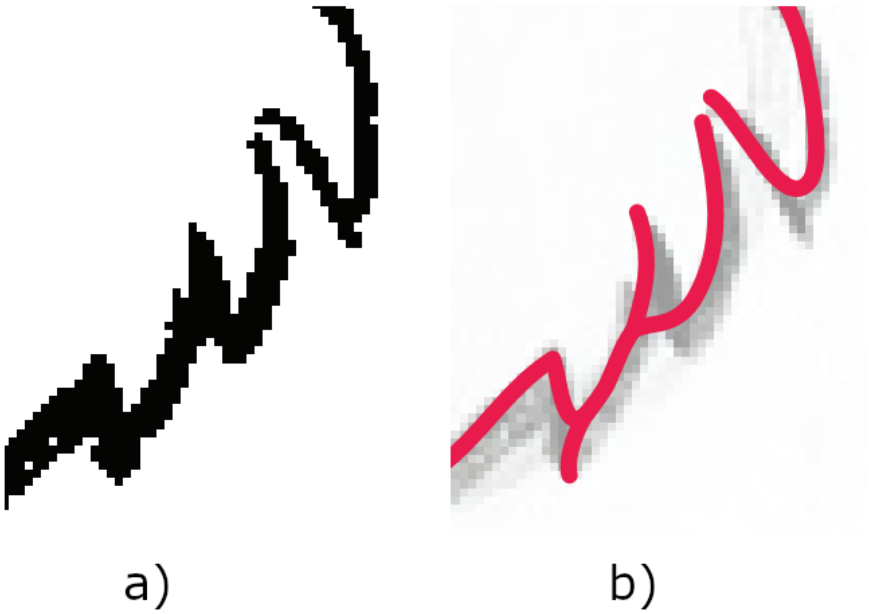
technicznych, nie będzie ona skuteczna przy rozpoznawaniu pisma odręcznego [3]. Co więcej, metody binaryzacji oparte na CNN wykorzystują funkcję ewaluacji na poziomie oceny klasyfikacji poszczególnych pikseli. To sprawia, że nie priorytetyzują one ciągłości linii. Możliwa jest więc sytuacja, że mimo zadowalającej oceny analitycznej, obraz zwróci wynik wizualnie nieakceptowalny. Innym problemem może być zbyt agresywne uproszczenie krawędzi, powodujące utratę informacji pożądanych w przypadku analizy pisma. Wektoryzacja ma na celu redukcję skomplikowania krzywych, co może doprowadzić do deformacji cech poszczególnych glifów. Taki efekt został przedstawiony na Rysunku 10. Można na nim zauważyć, że detale zostały wy-



Rysunek 10: Oryginalny obraz (po lewej) oraz efekt preprocessingu algorytmu wektoryzacji (po prawej)
Źródło: [27]

eliminowane. Warto zwrócić uwagę na okrągłe artefakty trzymane przez postać. Zawierają one strukturę dwóch okręgów o wspólnym środku. W wyniku algorytmu preprocessingu zostały one zredukowane do pojedynczej linii. Takie zachowanie jest nieakceptowalne w przypadku obrazów pisma.

W [12] zastosowano zwykłą binaryzację globalną w celu odseparowania pikseli tła. To powoduje powstanie wielu artefaktów, które w konsekwencji prowadzą do nieoptymalnego rozwiązania problemu wektoryzacji, który następnie musi zostać poprawiony ręcznie [12]. Przykład problematycznej linii został przedstawiony na Rysunku 11. Pojawiają się na nim klasyczne błędy powstałe wskutek binaryzacji. Są to ubytki w liniach, które można zaobserwować w lewym dolnym rogu Rysunku 11 a) oraz poszarpane krawędzie linii na całym jej obszarze. Kolejnym problemem jest przerwanie linii w prawej górnej części tego obrazu. Ponadto grubość linii jest nierównomierna. Problem ten wynika bezpośrednio z użycia binaryzacji globalnej, która ustala jeden próg niezależnie od intensywności. Dla tego problemu algorytm preprocessingu

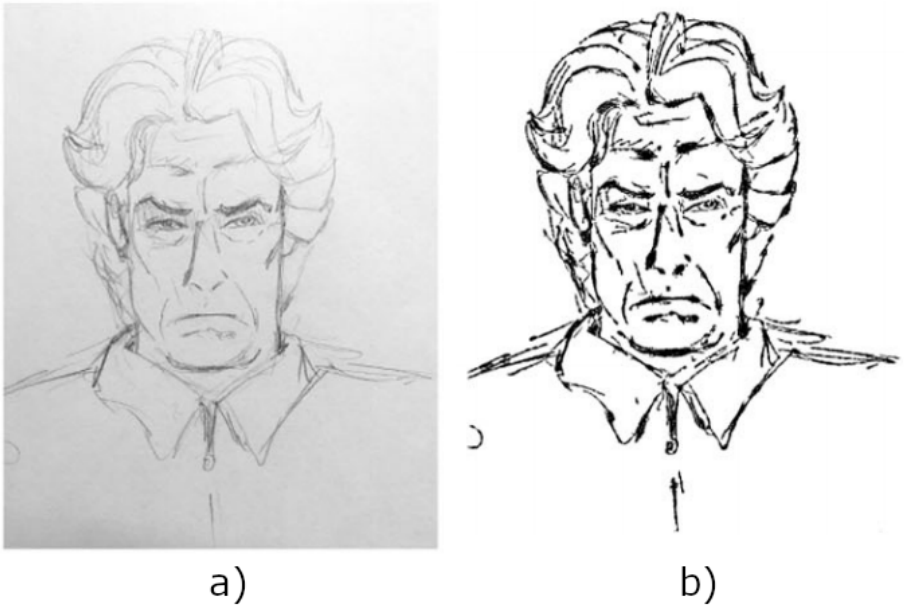


Rysunek 11: a) Obraz po binaryzacji
b) Obraz po automatycznej wektoryzacji, wymagającej ręcznych poprawek
Źródło: [12]

powinien wygenerować ciągłą linię o równej grubości. W praktyce rezultatem jest znacznie bardziej skomplikowana struktura, oznaczona kolorem czerwonym na Rysunku 11.

Aby wyekstrahować linię, Autorzy w [26] stosują korelację krzyżową z predefiniowanymi szablonami, na które następnie nakładają filtry górnoprzepustowe, filtr medianowy i binaryzację Otsu. Efekty takiego algorytmu pokazuje Rysunek 12. Zaprezentowany przykład obrazuje jeden z głównych problemów filtru medianowego. Potrafi on pogrubić obiekt. Na Rysunku 12 b), na nosie postaci oraz na krawędzi jej twarzy można zauważyć efekty takiego niepożądanego pogrubienia. Z drugiej strony nie wszystkie oczekiwane linie zostały połączone. Takie przypadki można zaobserwować we włosach postaci. Warto zauważyć, że binaryzacja Otsu generuje dużą ilość szumów. Filtr medianowy, podobnie jak filtry górnoprzepustowe powodują jedynie wzmacnianie tych szumów. Prezentowany przykład pokazuje kartkę równomiernie oświetloną, przez co problem nie jest tak mocno widoczny. Ponadto w zagadnieniu wirtualizacji rysunków uwzględnianie nawet najmniejszych artefaktów może być uzasadnione. Jednak w przypadku rozpoznawania pisma takie podejście byłoby niedopuszczalne z racji braku algorytmu, który niwelowałby liczbę niepoprawnie zaklasyfikowanych pikseli.

Problem rozpoznawania pisma jest często redukowany jedynie do klasycznego alfabetu ła-



Rysunek 12: a) Oryginalny obraz w odcieniach szarości
b) Obraz po binaryzacji i ścienianiu
Źródło: [26]

cińskiego zawierającego 26 znaków [58]. Warto mieć na uwadze, że algorytmy poruszające problem przetwarzania obrazu z pismem mogą okazać się znacznie mniej skuteczne w przypadku operowania na innych alfabetach. Co więcej, ich skuteczność może być znacznie niższa także dla alfabetu polskiego zawierającego znaki diakrytyczne. Język polski zawiera 9 liter wzbogaconych o znaki diakrytyczne. To sprawia, że problem rozpoznania pisma okazuje się być znacznie bardziej zaawansowanym, gdyż sporym wyzwaniem jest rozróżnienie następujących par liter: (a, ą), (c, ć), (e, ę), (l, ł), (n, ń), (o, ó), (s, ś), (z, ź), (z, ż), (ż, ź). Problem istnieje na każdym etapie procesu przetwarzania obrazu. Wszystkie etapy od preprocessingu, aż do klasyfikacji należy dostosować, by zwrócić szczególną uwagę na rozróżnianie podobnych do siebie glifów. W praktyce istnieje niewiele prac naukowych, skupiających się na tym zagadnieniu. Jednym z nich, jest ten zastosowany w [83]. Prezentuje on ciekawy pomysł transformacji każdego glifu do jego szkieletu. Zastosowano tu algorytm szkieletyzacji k3m [80], który następnie poddany został dylatacji. Dzięki temu możliwym było uwydatnienie znaków dialektycznych charakterystycznych dla języka polskiego. Przykładowy rezultat przetwarzania liter został pokazany na Rysunku 13. Jak można na nim zauważyć, algorytm powoduje znaczną deformację oryginalnych glifów. Mimo skuteczności uwydatniania cech charakterystycznych poszczególnych liter, ów algorytm powoduje ich deformację. To sprawia, że przedstawiony algorytm jest

Rysunek 13: Przykładowy rezultat preprocessingu oraz segmentacji liter alfabetu polskiego
Źródło: [83]

mało uniwersalny. Ma on zastosowanie w przypadku rozpoznania liter i próby odczytania zapisanej informacji. Jednak postawiony przed zadaniem identyfikacji osoby na podstawie pisma nie osiągnie wysokiej skuteczności. Zaakcentowanie pewnych cech liter oraz zignorowanie innych może doprowadzić do utraty unikalności cech właściciela pisma. Ponadto algorytm jest skoncentrowany stricte na rozpoznawaniu pisma i cech charakterystycznych dla niego.

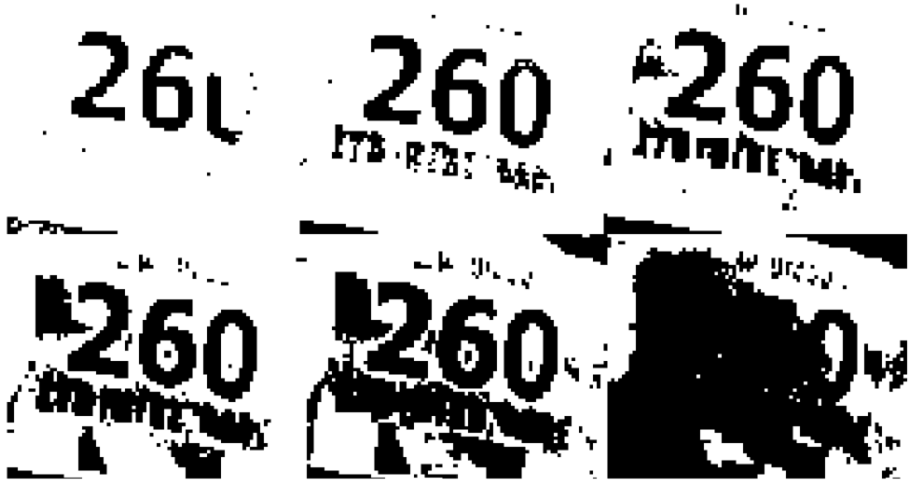
Jednym z podproblemów rozpoznawania tekstu jest zagadnienie rozpoznawania tablic rejestracyjnych pojazdów. Problem polega na identyfikacji prostokątnej tabliczki, na której znajduje się ciąg liter i cyfr. Głównym ułatwieniem w stosunku do zdjęć dokumentów jest fakt, że znaki na tablicy rejestracyjnej mają jedną, określoną, czytelną czcionkę. Nie występują tutaj artefakty związane z zagięciami, plamami, czy przebiciami tekstu z drugiej strony. Jednak głównym utrudnieniem w takim problemie jest często występujący efekt mocnego rozmycia zdjęcia. Fotografowanie tablicy rejestracyjnej pojazdu w trakcie ruchu może powodować także deformację znaków znajdujących się na niej. Dużo częstszym problemem jest także nierównomierne oświetlenie czy efekt flary. W ogólnym przeglądzie metod związanych z tym zagadnieniem [90] zauważono, że najskuteczniejszą metodą preprocessingu, mającą na celu wyeliminowanie szumów, jest zastosowanie filtru medianowego. Jest wiele algorytmów stosujących tę metodę preprocessingu [95, 44, 73]. Z kolei w [7] zaproponowano stosowanie filtru Gaussa. Jest to metoda o tyle niebezpieczna, że filtry dolnoprzepustowe, mimo możliwości usunięcia szumów, zwiększają rozmycie obrazu. Efekty takiego zabiegu przedstawia Rysunek 14. Można na nim dostrzec wiele postrzępionych linii. Kolejnym problemem jest niedokładne wypełnienie wszystkich glifów. Przykładowo, litera X zawiera w środku obszar czarny, należący do tła. Autorzy, po zastosowaniu filtru Gaussa i binaryzacji Otsu, zdecydowali się zastosować zamknięcie morfologiczne. Rezultaty pokazują, że nie jest to wystarczające podejście, by uzyskać znaki dobrze czytelne. Jednak zastosowany przez nich algorytm poprawnie zidentyfikował poszczególne

Rysunek 14: Przykładowy rezultat preprocessingu tablicy rejestracyjnej z wykorzystaniem jedynie filtru Gaussa, a następnie binaryzacji Otsu
Źródło: [7]

glify (oznaczone białymi ramkami na Rysunku 14). Takie rozwiązanie może być skuteczne w przypadku pisma maszynowego, jednak użyte do pisma odręcznego wyodrębnienie znaków przy tak dużej ilości szumów może okazać się niemożliwe. Ponadto filtr Gaussa zastosowany przez binaryzacją może powodować nieodwracalną utratę cech kluczowych do poprawnej identyfikacji pisma. W [90] poświęcono uwagę lokalnym metodom binaryzacji, które mogłyby zminimalizować czy nawet wyeliminować potrzebę wykorzystania metod preprocessingu. Najpopularniejszym algorytmem do postawionego problemu rozpoznawania tablic rejestracyjnych jest metoda Otsu [44, 7, 45], Niblack'a oraz Sauvoli [76]. Są to metody, które dobrze radzą sobie z nierównomiernym oświetleniem. Jednak, jak pokazują także testy w niniejszej pracy, w przypadku bardziej skomplikowanych obrazów stosowanie jedynie binaryzacji czy prostych filtrów nie jest wystarczającym rozwiązaniem. Zupełnie inne podejście zastosowano w [81]. Autorzy postanowili ograniczyć preprocessing jedynie do normalizacji rozdzielczości obrazu. Główną mechanikę identyfikacji znaków na tablicy rejestracyjnej przeniesiono na etap klasyfikacji, która polega na porównaniu kształtu poszczególnych znaków z przygotowanymi wzorcami. Liczba zgadzających się pikseli określa prawdopodobieństwo dopasowania. Takie podejście skutkuje skutecznością rozpoznawania tablic na poziomie 75%. Jest to rozczarowujący wynik, biorąc pod uwagę, że Autorzy wykorzystywali mało skomplikowane obrazy testowe. Ten przykład jednocześnie pokazuje, jak istotny jest etap preprocessingu w całym procesie przetwarzania obrazów.

Podobnym problemem jest rozpoznawanie numeru na koszulkach sportowców. Tego typu algorytmy mają za zadanie zidentyfikować numer na koszulce, twarz zawodnika, a następnie przypisać numer do twarzy. W tym przypadku problem rozpoznawania numeru na koszulkach jest zredukowany jedynie do cyfr, co znacznie ułatwia proces klasyfikacji, gdyż klas jest jedynie dziesięć (dla każdej z cyfr 0-9). Co więcej, analogicznie do problemu tablic rejestracyjnych, numery na plastronie z założenia mają za zadanie być dobrze widoczne, więc są wydrukowane

czcionką komputerową. Może się więc wydawać, że preprocessing nie jest w tym przypadku istotny, z racji tak uproszczonego problemu. Jednak pojawiają się tu problemy analogiczne jak przy rozpoznawaniu tablic rejestracyjnych. Problemy wynikają nie tylko z potencjalnego rozmycia zdjęcia. Jako że fotografowany sportowiec często jest w ruchu, pojawia się problem zagiętej czy zasłoniętej części numeru. Rolą preprocessingu jest jak najdokładniejsze odwzorowanie widocznej części numeru. W [105] zrezygnowano z klasycznego podejścia do binaryzacji. Dokonano kwantyzacji kolorów [16], w wyniku czego wyjściowy obraz posiadał ograniczoną liczbę n kolorów. Następnie zastosowano $n-1$ binaryzacji globalnych. Efekt takiego podejścia przedstawia Rysunek 15. Na każdym z tak powstałych obrazów przetworzono powstałe



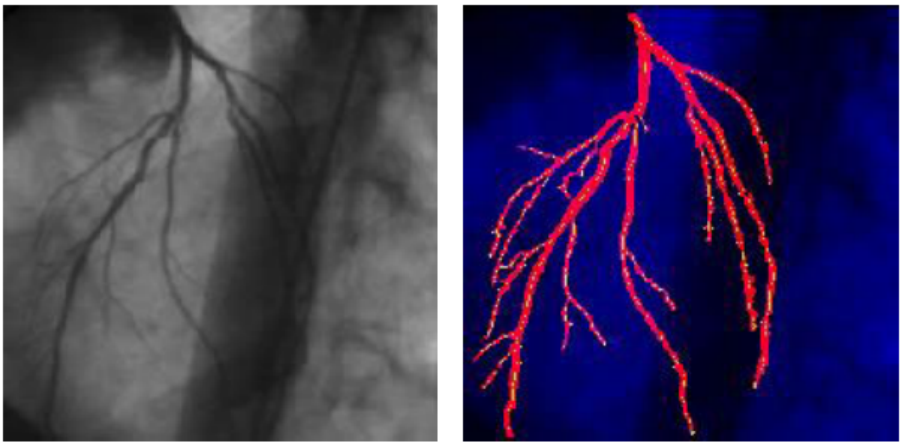
Rysunek 15: Sześciokrotna binaryzacja obrazu z numerem na koszulce sportowca, dokonana po kwantyzacji kolorów
Źródło: [105]

kontury. Takie podejście okazuje się być skuteczne do rozpoznawania znaków przy odpowiednich założeniach. Przykładowo, Autorzy zakładają, że szukane znaki mają wysokość większą niż szerokość. Jest to poprawne założenie w przypadku operowania jedynie na cyfrach. Jednak przy próbie identyfikacji dowolnego glifu takie podejście nie wystarcza. Pojawia się problem rozróżnienia obszaru obiektu od szumu, który powinien zostać usunięty. Ponadto kilkukrotna binaryzacja może zwrócić różne efekty dla tego samego glifu. Kolejnym problemem jest wydajność algorytmu. Powtarzanie tych samych operacji dla każdego z obrazów wydłuża czas działania algorytmu n-krotnie. Próba konstrukcji jednego obrazu wyjściowego na podstawie różnych danych jest zagadnieniem znacznie trudniejszym w przypadku pisma odręcznego. Autorzy w [108] podjęli podobną próbę rozpoznawania numerów na koszulkach piłkarzy. Także

zastosowali algorytm kwantyzacji kolorów. Następnie zdecydowali się na operację otwarcia morfologicznego, co okazało się być wystarczającą metodą, by wyeliminować szumy z przetwarzanych obrazów. Jest to podejście warte przetestowania. Taka operacja pozwala wyeliminować niewielkie artefakty z obrazów. Warto zauważyć, że numery na koszulkach sportowców charakteryzują się znaczną grubością. To sprawia, że otwarcie morfologiczne nie wpływa negatywnie na ciągłość numerów. W przypadku pisma odręcznego linia pisma jest zazwyczaj znacznie cieńsza. Zastosowanie otwarcia na takich obrazach może powodować utratę ciągłości linii, zwłaszcza w przypadku, gdy zdjęcie zawiera szumy, które dodatkowo utrudniają utrzymanie ciągłości.

Przetwarzanie obrazów naczyń krwionośnych z pozoru wydaje się być zupełnie innym problemem w porównaniu do przedstawionych powyżej. Jednak w praktyce w obu przypadkach algorytmy preprocessingu skupiają się na wydobyciu struktur liniowych z obrazów, przy zachowaniu ciągłości oraz ignorowaniu istniejącego szumu. W przypadku obrazów medycznych, proces ich przetwarzania często nie kończy się na klasyfikacji, lecz jedynie na poprawieniu ich jakości i podkreśleniu pewnych kluczowych cech. Tak przygotowany obraz jest następnie analizowany przez wykwalifikowanego specjalistę, który podejmuje decyzję. To sprawia, że priorytetem dla filtrów skupiających się na problematyce medycznej jest podkreślenie pikseli obiektu - naczyń krwionośnych. W [49] przedstawiono przegląd algorytmów analizujących obrazy medyczne. Badania wskazują na to, że większość z nich wymaga interakcji z użytkownikiem. W szczególności specjalista zaznacza interesujące go elementy obrazu przy pomocy serii kliknięć. Świadczy to o znacznym skomplikowaniu takich zdjęć. Przykładowe zdjęcie medyczne zostało przedstawione na Rysunku 16. Z drugiej strony istnieją także algorytmy nie wymagające tej interakcji, a zamiast tego mają dostęp do wiedzy a priori [72]. Kolejną istotną obserwacją jest fakt, że większość algorytmów klasyfikujących choroby przedstawione w [49] nie wykorzystuje algorytmów preprocessingu. To może tłumaczyć niską skuteczność takich metod i konieczność korzystania z zewnętrznej informacji.

Z kolei metody preprocessingu wiążą się ze zmodyfikowaniem obrazu wejściowego, więc niektóre jego fragmenty mogą wyglądać inaczej niż na obrazie źródłowym. Zazwyczaj, w wyniku stosowania takich metod na obrazach medycznych ostateczny obraz mocno pogrubia obiekt znajdujący się na nim (np. naczynia krwionośne). W porównaniu do obrazu wejściowego są one bardziej czytelne i praktyczniejsze z punktu widzenia podejmowania decyzji przez lekarza. Jednak patrząc na problem pod kątem dokładności odwzorowania obrazu wejściowego,



Rysunek 16: Zdjęcie naczyń krwionośnych (po lewej) oraz wysegmentowana istotna część obrazu (po prawej, oznaczona kolorem czerwonym)
Źródło: [49]

należy uwzględnić te różnice. Wyjściowy obraz może reprezentować strukturę większą niż w rzeczywistości. Co więcej, proporcje między poszczególnymi regionami na obrazie mogą nie odzwierciedlać rzeczywistych wymiarów. Specjalista analizujący taki obraz powinien brać te modyfikacje pod uwagę i być ich świadomy. Przykładem medycznego algorytmu preprocessingu jest filtr Frangi'ego [31]. Jest to uniwersalna metoda identyfikacji krawędzi obiektów, w szczególności wykorzystywana podczas identyfikacji naczyń krwionośnych. Efekty jego działania pokazuje Rysunek 17. Do obliczania kierunków wyznaczany jest hesjan w poszczególnych regionach obrazu. Struktura powierzchni jest przewidywana na podstawie wartości własnych. Decyzja o użyciu operacji morfologicznych następuje na podstawie znaków tych kierunków. Algorytm cechuje się dużą odpornością na szumy, co jest szczególnie istotne dla obrazów medycznych. Z racji specyfiki problemu, Autorzy algorytmu nie proponują analitycznej metody oceny rezultatów działania swojej metody. Ocena wizualna pozwala przypuszczać, że zastosowanie algorytmu do pisma powinno dawać zadowalające efekty. Główną niepewnością związaną z jego stosowaniem jest typowy problem filtrów medycznych, którym jest powiększanie objętości obiektów. Inny przykład prezentuje filtr przetwarzający naczynia krwionośne w gałkach ocznych [107]. Autorzy zastosowali ciekawy sposób redukcji szumu. Zauważyli, że kanał zielony z modelu przestrzeni barw RGB oferuje najlepsze rozróżnienie pikseli obiektu od pikseli tła. Wobec tego obraz w odcieniach szarości jest otrzymywany z obrazu kolorowego poprzez uwzględnienie intensywności w jednym kanale. Następnie, po binaryzacji lokalnej, na obrazie stosowany jest filtr medianowy. Metoda z wyborem jednego kanału, na którym są wykony-

Rysunek 17: Obraz medyczny w odcieniach szarości (po lewej) oraz zidentyfikowane naczynia krwionośne na tym obrazie (po prawej)
Źródło: [31]

wane dalsze operacje, jest specyficzna dla tego problemu. Nie znajdzie więc zastosowania do obrazów z pismem odręcznym. Wymaga także dostępu do kolorowego obrazu, co jest dodatkowym założeniem do poprawnego działania algorytmu. Sam filtr medianowy okazuje się być wykorzystywany zarówno do problemów pisma, jak i do problemów medycznych. Jest to metoda prosta i uniwersalna, jednak nie uwzględniająca kierunku krawędzi. Stosowanie jednej, zazwyczaj symetrycznej maski, okazuje się dawać zadowalające wyniki. Należy zaznaczyć, że dostosowanie owej maski do aktualnie przetwarzanego regionu mogłoby polepszyć efekt końcowy i wyeliminować niepożądane zwiększenie objętości przetwarzanych obiektów.

Warto zauważyć, że preprocessing jest etapem przetwarzania obrazu, którego efekty silnie zależą od bazy zdjęć, na których algorytm jest testowany. Dobór odpowiedniego, zróżnicowanego zestawu jest kluczowy, by ocenić sposób działania algorytmu oraz zidentyfikować jego zalety i słabe strony. W przypadku korzystania z czarno-białych, uprzednio przygotowanych skanów zdjęć, eliminowany jest problem związany z niedokładnością narzędzi pomiarowych i z kiepską jakością fotografowanych danych. Istnieje wiele algorytmów rozpoznających pismo ręczne na skanach dokumentów i dających obiecujące wyniki [60, 51, 6]. Mimo to, rzeczywista skuteczność tych metod jest nieznana, póki ich zachowanie nie zostanie przetestowane na rzeczywistych, zaszumionych zdjęciach. Jedną z klasycznych baz używanych do analizy pisma

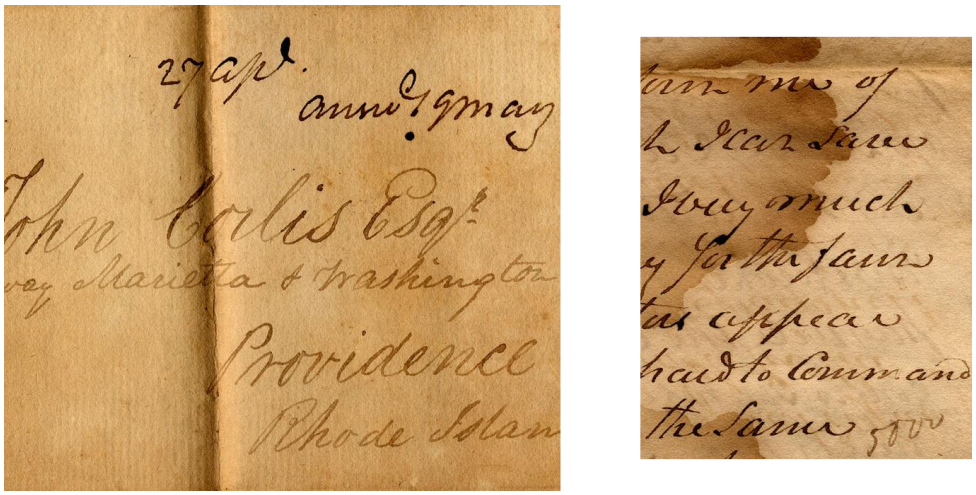
odręcznego jest baza MNIST [52]. Jest ona szeroko wykorzystywana do testowania algorytmów klasyfikacji. Niestety, jest ona bezużyteczna z punktu widzenia algorytmów preprocessingu, gdyż zawiera ona obrazy zbinaryzowane i pozbawione szumów. Przykładowe obrazy przedstawiono na Rysunku 18. Jest to baza ignorująca jeden z głównych problemów podczas

Rysunek 18: Przykładowe obrazy w bazie MNIST

procesowania obrazu, taki jak niedostatecznie dobra jakość obrazów wejściowych. Literatura pokazuje, że klasyfikatory weryfikowane na niej osiągają niemal bezbłędny wynik, jak np. skuteczność na poziomie 99.03% algorytmu opartego na sieciach konwolucyjnych [74]. Baza była używana przez wiele zróżnicowanych klasyfikatorów, jak oparte na SVM [34] czy sieciach głębokich [71, 19]. Stosowanie algorytmów preprocessingu do takiej bazy mija się z celem, gdyż ich wpływ na ostateczny rezultat jest znikomy, tym samym ciężko ocenić ich jakość i porównać między sobą.

Znacznie lepszymi bazami są takie, które zawierają obrazy pełne niedoskonałości. W szczególności stare zdjęcia, o niskiej rozdzielczości czy rozmazane są interesujące z punktu widzenia preprocessingu. Obrazy tego typu są z kolei gorsze do porównania w klasyfikacji. Znaczącym problemem przy takich zdjęciach jest sam proces segmentacji oraz ekstrakcji cech, które są bardzo wrażliwe na wszelkie zmiany w etapie preprocessingu. Takie obrazy skupiają ciężar procesu rozpoznawania obiektów na obrazie na wcześniejszych etapach, gdzie znacznie większym problemem jest ekstrakcja cech niż podział ich na ustalone klasy. Tym samym są to najbardziej pożądane obrazy z punktu widzenia oceny rezultatów preprocessingu.

Przykładem baz danych zawierającym omówione obrazy jest baza DIBCO [62]. Przykładowe obrazy z tej bazy zostały przedstawione na Rysunku 19. Mimo że obrazy prezentują tekst, semantyka czy rozpoznanie poszczególnych glifów nie jest istotne. To, co wysuwa się na pierwszy plan podczas analizowania obrazów tej bazy to zagięcia, plamy czy wynik niedos-

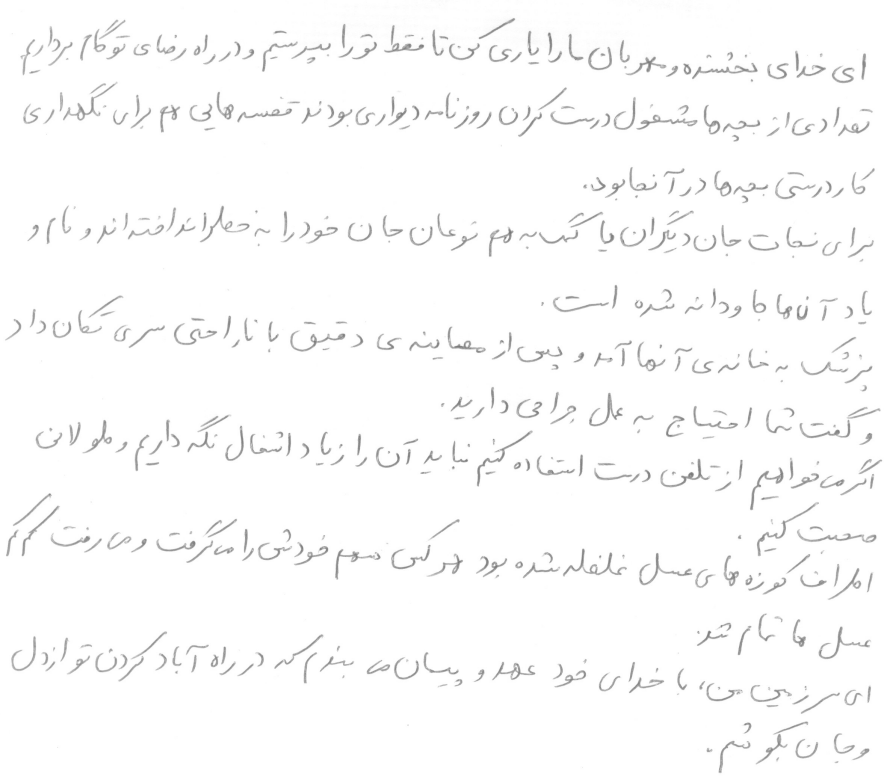


Rysunek 19: Dwa przykładowe obrazy bazy DIBCO

konałości papieru, który został wykorzystany. Efektem stosowania papieru o niskiej nieprzezroczystości drukowej może być przebijanie tuszu z drugiej strony. Taki efekt powinien zostać uznany za szum i wyeliminowany w metodach postbinaryzacyjnych. Warto zwrócić uwagę, że baza przedstawia zdjęcia spotykane w praktycznych sytuacjach, z którymi może obcować użytkownik aparatu. Są to przypadki nie uwzględniane przez większość baz, które są skupione na problemie klasyfikacji. Tym samym omijają one bardzo istotny problem obrazów kiepskiej jakości. Przedstawiona baza DIBCO skupia się na preprocessingu. Do każdego oryginalnego obrazu przypisany jest obraz wzorcowy, przedstawiający idealny obraz po procesie binaryzacji. To umożliwia zaprojektowanie miar skuteczności i tym samym porównać algorytmy preprocessingu między sobą. Sposoby weryfikacji wyników z użyciem obrazów wzorcowych zostały szczegółowo opisane w rozdziale 4. Warto zwrócić uwagę na fakt, że dzięki obrazom wzorcowym zweryfikowany może zostać jedynie etap preprocessingu (widoczny na schemacie na Rysunku 2). Pozwala wyeliminować wpływ pozostałych etapów (segmentacja, ekstrakcja cech, klasyfikacja) na efekty testów. Dzięki temu pozwalają one na uzyskanie bardziej wiarygodnej oceny pojedynczego etapu. Baza danych posłuży przede wszystkim do ustalenia, w jaki sposób algorytm reaguje na losowe utrudnienia. Jest to także baza często spotykana w literaturze do oceny metody preprocessingu. To daje możliwość obiektywnego porównania wyników różnych algorytmów. Z racji zróżnicowania problemów na tych obrazach, nawet najskuteczniejsze algorytmy mogą nie poradzić sobie z przetworzeniem poszczególnych przypadków, co ułatwia przeprowadzenie analizy w zakresie porównania różnych rozwiązań pod kątem zalet i wad.

Kolejną bazą wartą analizy pod kątem preprocessingu jest PBOK [5]. Zawiera ona obrazy

pisma perskiego. Przykład takiego obrazu został przedstawiony na Rysunku 20. Rodzaj pi-



Rysunek 20: Przykładowy obraz bazy PBOK

sma może mieć znaczenie w przypadku niektórych algorytmów preprocessingu. Przykładem może być algorytm bazujący na sieciach konwolucyjnych, który był uczony na alfabecie łacińskim [104]. Przedstawiona baza jest znacznie prostsza od DIBCO. Trudność w tym przypadku objawia się w kształtach linii, różnych od klasycznego problemu z wykorzystaniem alfabetu łacińskiego. Baza pozwala określić, jak poszczególne algorytmy preprocessingu radzą sobie z mniej typowym problemem. Omawiana baza nie zawiera zdjęć z plamami czy zagnieceniami, więc powinna być relatywnie prostsza w stosunku do DIBCO. Jednak problemem w tym przypadku jest wyblakły, rozmywający się tekst. To sprawia, że algorytmy są bardziej podatne na problem przerwanej ciągłości linii [28]. Ponadto język perski posiada wiele akcentów, które są reprezentowane jako niewielkie linie bądź kropki nad poszczególnymi glifami. Istnieje niebezpieczeństwo, że takie małe obiekty zostaną usunięte razem z szumem powstałym w trakcie binaryzacji. Jest to niepożądany efekt. Baza PBOK posłuży także do zweryfikowania, jak algorytmy zachowują się z mniej typowym problemem. W przypadku alfabetu łacińskiego wiele algorytmów jest dostosowanych pod konkretną grupę glifów. Przypadek mniej popularnego alfabetu pozwoli zbadać uniwersalność metody. Będzie to także wskazówka, czy dany algorytm

byłby skuteczny dla alfabetu polskiego, zawierającego znaki dialektyczne.

3 Opis zastosowanych technik

W niniejszym dziale przedstawione zostaną szczegóły metody, której pomysłodawcą był Marcin Adamski [82], a także rezultaty zaimplementowanego przeze mnie algorytmu i eksperymenty udowadniające jego skuteczność.

3.1 Zarys zaproponowanej metody

Na wejściu, metoda musi mieć dostęp do obrazu w odcieniach szarości. Algorytm nie precyzuje metody binaryzacji, której należy użyć. Działa poprawnie z różnymi metodami, co zostanie wykazane w niniejszej pracy.

Równolegle do procesu binaryzacji, algorytm oblicza pole kierunkowe, wykorzystując do tego obraz w odcieniach szarości. Niniejsza praca nie ogranicza się do jednej metody wyznaczania pola. Przedstawia i porównuje dwa podejścia. Istotnym czynnikiem dla algorytmów wyznaczających pole kierunkowe jest ich odporność na szum obrazu wejściowego.

Pierwszy algorytm obliczający to pole został zaprezentowany w [38]. Jest to podejście oparte o gradient, pierwotnie wykorzystywane do obrazów zawierających linie papilarne. Ponadto brane są pod uwagę możliwe niedokładności w jego działaniu, a rzetelność zwracanych rezultatów jest dodatkowo oceniana przez miarę koherencji zaprezentowaną w [11]. Kolejnym usprawnieniem jest dodanie analizy histogramu kołowego i dokładniejsza analiza kierunków w regionach obiektu o niskiej wartości miary koherencji.

Druga metoda to podejście oparte o hesjan. Zaproponował je Frangi w [31] w celu przetwarzania obrazów medycznych zawierających naczynia krwionośne. Rezultaty przedstawione w [9] wskazują, że algorytm wyróżnia się odpornością na szum.

Kolejnym etapem procesowania prezentowanego w niniejszej pracy algorytmu jest zastosowanie operacji morfologicznych. Zastosowana dylatacja charakteryzuje się adaptacją elementu strukturalnego do aktualnie przetwarzanego regionu obrazu, zależnie od pola kierunkowego. W przypadku metody opartej na gradiencie, uwzględniana jest również miara koherencji. Jeśli jej wartość przekracza ustalony próg, algorytm wykorzystuje liniowy element strukturalny z kierunkiem zależnym od pola kierunkowego. W przeciwnym przypadku wartość pola kierunkowego uznawana jest za nierzetelną. Wówczas zostały zaprezentowane dwa podejścia postępowania: zastosowanie elementu strukturalnego w kształcie diamentu lub dalsza analiza regionu, bazująca na histogramie kołowym, w celu identyfikacji wielu kierunków.

Ostatnim krokiem algorytmu jest zastosowanie testu 4-sąsiedztwa. Pozwala on usunąć z obrazu niewielkie artefakty pozostałe po poprzednich operacjach morfologicznych.

Pełen schemat działania algorytmu został przedstawiony na schemacie na Rysunku 21. Poszczególne wiersze przedstawiają kolejne etapy działania algorytmu. Należy tutaj podkreślić elastyczność algorytmu. Elementy w każdym wierszu można stosować zamiennie. Ponadto niektóre z etapów można pominąć. Przykładowo decyzja, czy uwzględnić skalowanie lub test 4-sąsiedztwa należy do osoby projektującej rozwiązanie. Najczęściej wpływ na decyzję ma złożoność problemu - jeśli pozostałe etapy dają zadowalające efekty, można zrezygnować z tych opcjonalnych. Warto mieć na uwadze, że każdy etap polepsza rezultat końcowy, ale jednocześnie wpływa negatywnie na szybkość przetwarzania pojedynczego obrazu. W przypadku gdy to szybkość działania jest priorytetem, najlepszym wyborem będzie zrezygnowanie z opcjonalnych etapów. Złożoność obliczeń zależy także od rozdzielczości obrazów wejściowych. Kolejnym rozwiązaniem na zaoszczędzenie czasu jest zmniejszenie rozmiaru zdjęcia.

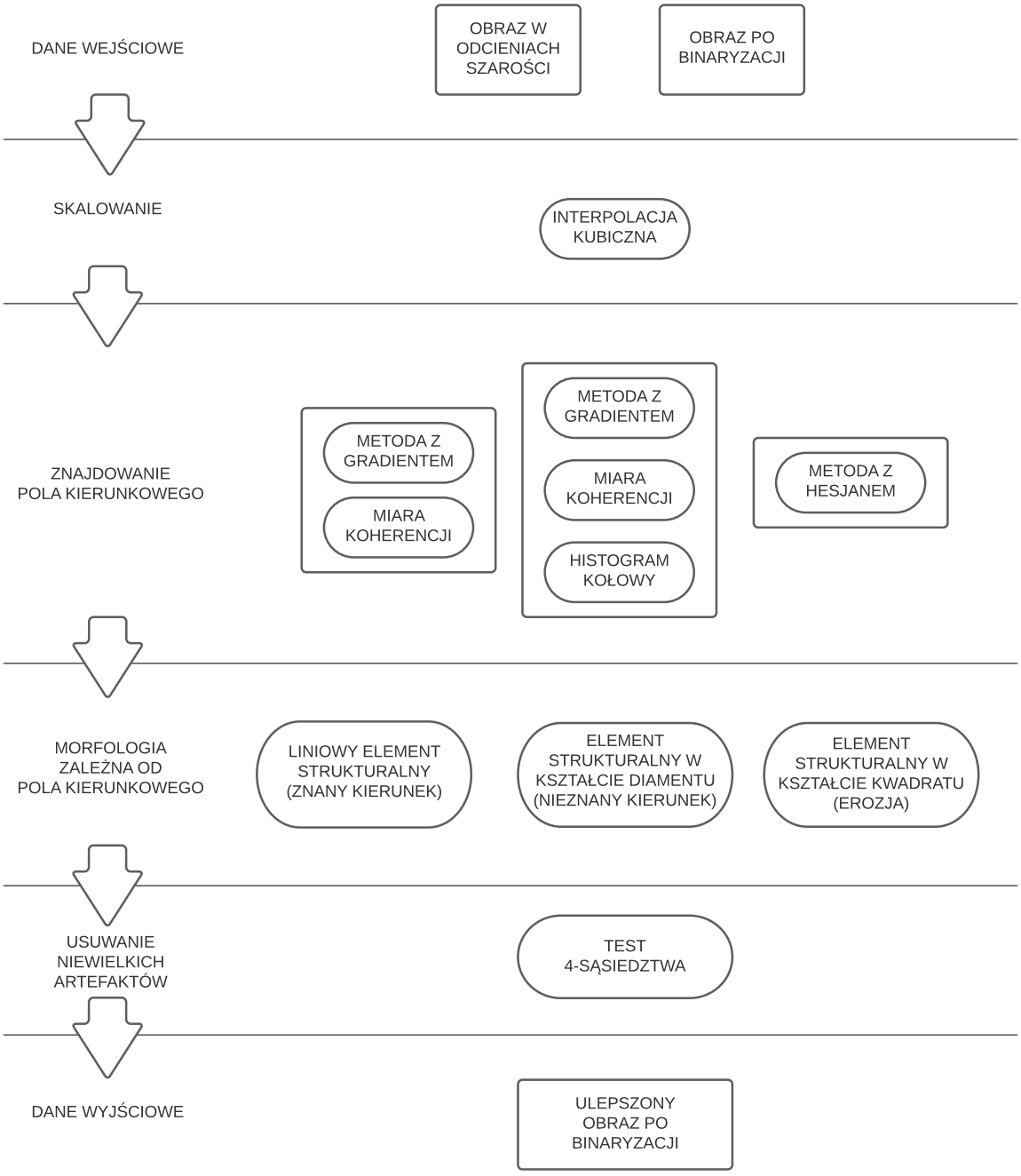
3.2 Binaryzacja

Binaryzacja jest procesem konwertującym obraz wejściowy, z reguły w odcieniach szarości, na obraz binarny. Celem tej operacji jest segmentacja pikseli obiektu od pikseli tła. Literatura proponuje wiele zróżnicowanych podejść do binaryzacji. W szczególności na uwagę zasługuje pozycja prezentująca ponad 40 metod [87], natomiast najnowsze postępy są przedstawione w [17, 37, 42, 103, 104]. Mnogość metod skutkuje tym, że w zależności od zadania, inny typ binaryzacji może okazać się najwydajniejszy.

Binaryzację można zdefiniować jako proces przekształcania obrazu w odcieniach szarości (bądź kolorowego) do obrazu binarnego. Każdy piksel zostaje poddany testowi: jeśli wartość jego intensywności jest mniejsza od progu, zostaje zaklasyfikowany jako tło. W przeciwnym przypadku jest uznawany za piksel należący do obiektu. Ta klasyfikacja została opisana poprzez równanie (1).

$$g(x,y) = \begin{cases} 0 & \texttt{, jeśli } g(x,y) < T \\ 1 & \texttt{, w przeciwnym przypadku} \end{cases} \tag{1}$$

T w tym równaniu jest progiem. Jeśli jest to liczba stała dla każdego piksela na obrazie, taki algorytm jest określany binaryzacją globalną. W przeciwnym przypadku, tzn. jeśli próg jest obliczany dla każdego piksela osobno, taki algorytm nosi miano binaryzacji lokalnej. Mimo dłuższego czasu działania, jest ona częściej stosowana od binaryzacji globalnej i zazwyczaj



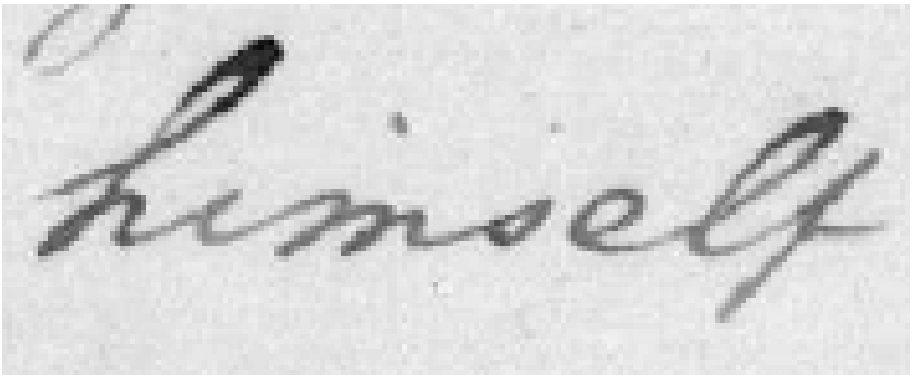
Rysunek 21: Schemat działania proponowanej metody

daje znacznie lepsze efekty.

Dostępne są także modyfikacje klasycznego podejścia do binaryzacji, przykładowo wielopoziomowa binaryzacja (dzieląca piksele na trzy lub więcej kategorii) została zaprezentowana w [59]. Celem takiego zabiegu jest osiągnięcie balansu między obrazem w odcieniach szarości i zbinaryzowanym. Ten pierwszy zawiera dużo informacji, z kolei jest trudny do procesowania, z racji braku określenia, które piksele należą do tła, a które do obiektu. Z kolei obraz po binary-

zacji ma klarownie rozdzielone obie grupy pikseli. Jednak ilość informacji na takim obrazie jest znacznie ograniczona. Wielopoziomowa binaryzacja, jeśli zostanie poprawnie wykorzystana, może łączyć zalety obu podejść.

Algorytm zastosowany w niniejszej pracy został przetestowany z następującymi algorytmami binaryzacji: Otsu [69], Sauvoli [84] oraz Niblacka [68]. Algorytmy zostały użyte na obrazie przedstawionym na Rysunku 22, dla którego istnieje także oczekiwany obraz po wzorcowej binaryzacji.



Rysunek 22: Wzorcowy obraz, na którym będzie dokonywana binaryzacja różnymi metodami

3.2.1 Metoda Otsu

Metoda Otsu [69] jest jedną z najbardziej popularnych metod binaryzacji. Wartość progu jest obliczana poprzez maksymalizację wariancji σ_B^2 między dwiema klasami pikseli, które reprezentują obiekt oraz tło. Można ją obliczyć, wykorzystując równanie (2).

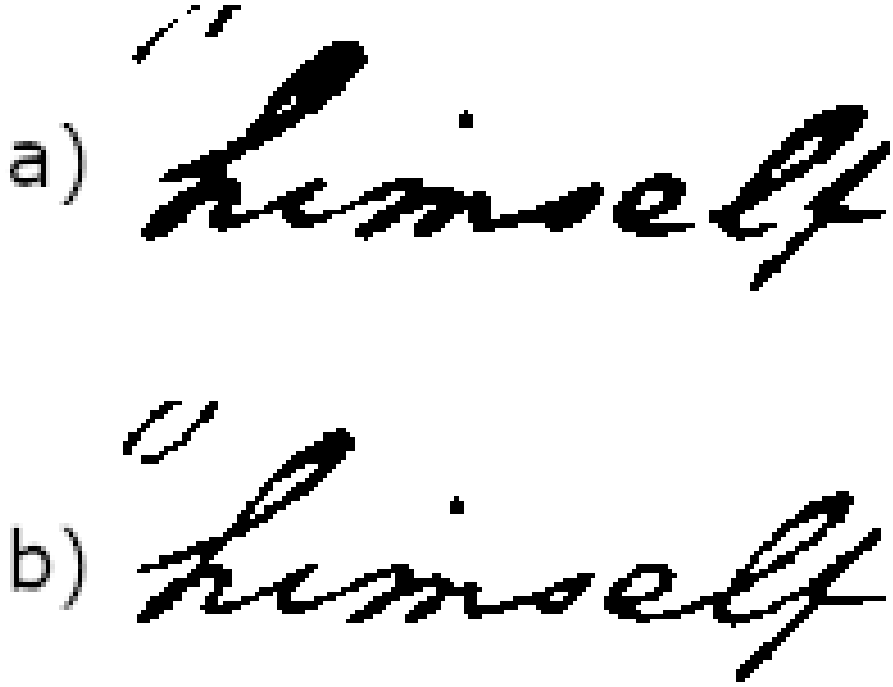
$$\sigma_B^2 = P_0 P_1 (\mu_0 - \mu_1)^2 \tag{2}$$

Gdzie μ_i jest średnią pikseli obrazu wejściowego, które zostały zaklasyfikowane jako obiekt ($i = 0$) oraz jako tło ($i = 1$). P_i jest sumą wartości histogramu po normalizacji, dla każdej klasy. Jeśli L jest liczbą intensywności na obrazie, próg globalny może zostać wyliczony w sposób przedstawiony w równaniu (3).

$$T_{opt} = \arg \max_{0<t<L-1} \sigma_B^2(t) \tag{3}$$

W swojej podstawowej formie, metoda Otsu wykorzystuje globalny, pojedynczy próg dla całego obrazu. To może skutkować niepoprawną segmentacją obrazów oświetlonych nierównomiernie lub zawierających dużą ilość szumu. W takich sytuacjach rozwiązaniem jest zasto-

sowanie Otsu z partycjonowaniem [17]. Obraz jest początkowo podzielony na sekcje z wykorzystaniem histogramu bimodalnego. Przykładowy efekt binaryzacji metodą Otsu został przedstawiony na Rysunku 23. Można na nim zauważyć, że metoda Otsu zacznie pogrubia linie



Rysunek 23: a) Obraz poddany binaryzacji metodą Otsu
b) Oczekiwany rezultat binaryzacji
Źródło: Opracowanie własne

pisma. Niestety, nie wiąże się to z gwarancją zachowania ciągłości linii. W lewym górnym rogu obu obrazów znajduje się fragment tekstu, który został niemal w połowie wyeliminowany po binaryzacji. Dużym problemem przy tej metodzie jest także zamykanie się pętli. Jest to skutek pogrubienia obiektów. Litery h oraz s wyrazu $himself$ na Rysunku 23 pokazują utratę tej istotnej informacji. Jest to szczególnie problematyczne w przypadku litery s, która została całkowicie zdeformowana próba ekstrakcji jej cech i dalszej poprawnej klasyfikacji może okazać się niemożliwa, niezależnie od przyjętego algorytmu postbinaryzacji. Warto zwrócić uwagę na literę h. Na oryginalnym obrazie w odcieniach szarości (Rysunek 22) intensywność koloru tuszu różni się dla górnej oraz dolnej części litery. Metoda Otsu zgodnie z oczekiwaniami uznała obie części za jeden spójny obiekt. Dzięki temu litera h nie posiada zbędnych przerw.

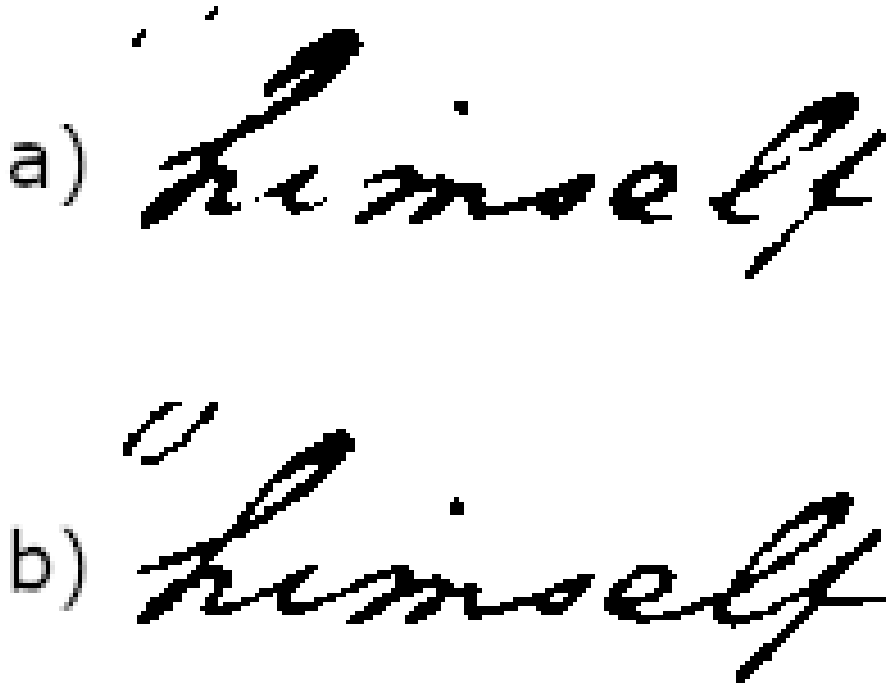
3.2.2 Metoda Sauvoli

W algorytmie Sauvoli [84] stosowany jest próg lokalny - wartość progu jest obliczana dla każdego piksela obrazu wejściowego, zgodnie z równaniem (4).

$$T(x,y) = \mu(x,y)\left[1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right] \tag{4}$$

gdzie $\mu(x,y)$ jest średnią, $\sigma(x,y)$ jest odchyleniem standardowym w otoczeniu piksela (x,y), R zakresem odchylenia standardowego, k jest parametrem zdefiniowanym przez użytkownika.

Rozmiar okna (określającego sąsiedztwo piksela (x,y)) oraz wartość parametru k wybierane są heurystycznie poprzez analizę przetwarzanego zbioru obrazów. W prezentowanych eksperymentach użyto rozmiaru 15×15 oraz $k = 0,5$, zgodnie z propozycją zawartą w [84]. Rezultaty zastosowania tego typu binaryzacji zostały przedstawione na Rysunku 24. W porównaniu do



Rysunek 24: a) Obraz poddany binaryzacji metodą Sauvoli
b) Oczekiwany rezultat binaryzacji
Źródło: Opracowanie własne

metody Otsu otrzymane pismo ma cieńsze linie. To generuje nowy problem - przerwania pętli. Przykłady takich przerwań można zaobserwować w literach h oraz l wyrazu $himself$. W szczególności litera h cechująca się różną intensywnością tuszu na obrazie wejściowym 22 jest

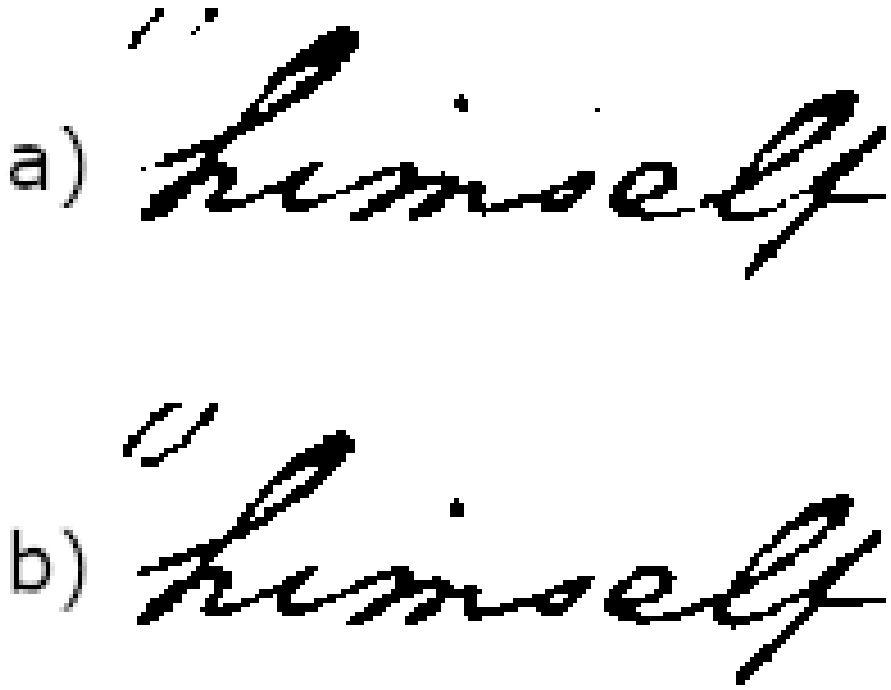
przetworzona gorzej w porównaniu do metody Otsu. Warto zwrócić uwagę na lewą część pętli litery. Ma ona dużą grubość, po czym nagle ulega zakończeniu. Miejsce zakończenia linii pokrywa się z miejscem nagłej zmiany intensywności tuszu na obrazie w odcieniach szarości. Kolejnym problemem jest utrata ciągłości między poszczególnymi glifami. Obraz wzorcowy zawiera połączenia między parami: (h, i), (e, l), (l, f). Wynik binaryzacji metodą Sauvoli nie takich połączeń nie posiada. Z punktu widzenia problemu rozpoznawania pisma, procesem zazwyczaj zachodzącym podczas segmentacji jest odseparowanie poszczególnych znaków od siebie. Wobec tego nasuwa się wniosek, że utrata połączenia między literami jest zjawiskiem pożądanym. Mimo to, algorytm preprocessingu ma za zadanie jak najdokładniej odwzorować cechy obrazu wejściowego i brak informacji o połączeniu należy traktować jako błąd.

3.2.3 Metoda Niblacka

Podobnie do metody Sauvoli, wartość progu w tej metodzie jest również obliczana lokalnie, zgodnie z poniższym równaniem:

$$T(x, y) = \mu(x, y) + k * \sigma(x, y) \tag{5}$$

Wartość średnia $\mu(x, y)$ oraz odchylenie standardowe $\sigma(x, y)$ są obliczone wewnątrz okna o stałym rozmiarze, z kolei k jest parametrem dobieranym przez użytkownika. W eksperymentach przedstawionych w niniejszej rozprawie zostało użyte okno o wymiarach 15×15. Przykładowy rezultat użycia tej metody binaryzacji został przedstawiony na Rysunku 25. Należy zwrócić uwagę na grubość linii. Jest ona cieńsza zarówno od wyniku zwróconego przez binaryzację Otsu, jak i Sauvoli. Tym samym najlepiej odwzorowuje obraz wzorcowy w tym przypadku. Algorytm najlepiej poradził sobie z pętlami wewnątrz liter h i l w wyrazie $himself$. W szczególności w przypadku litery h algorytm nie wypełnił tak dużej powierzchni pętli pikselami obiektu, jak zrobiły to pozostałe omawiane metody. Kolejną wartą odnotowania obserwacją jest litera s. Piksele tła nie zostały wyeliminowane z jej wnętrza, co jest dużym postępem w porównaniu do metod Otsu i Sauvoli. Mimo mniejszej grubości linii, metoda Niblacka dobrze sprawdziła się w utrzymaniu ciągłości między poszczególnymi literami. Jednak warto zauważyć, że istnieją niewielkie przerwy między literami e oraz l, a także l oraz f, czyniąc je osobnymi obiektami. To świadczy o potrzebie zastosowania metod postbinaryzacji, które powinny naprawić ten problem. Binaryzacja Niblacka jest bardziej wrażliwa na szum. W porównaniu do pozostałych metod, jako jedyna zaklasyfikowała artefakt nad literą s jako obiekt.



Rysunek 25: a) Obraz poddany binaryzacji metodą Niblacka
b) Oczekiwany rezultat binaryzacji
Źródło: Opracowanie własne

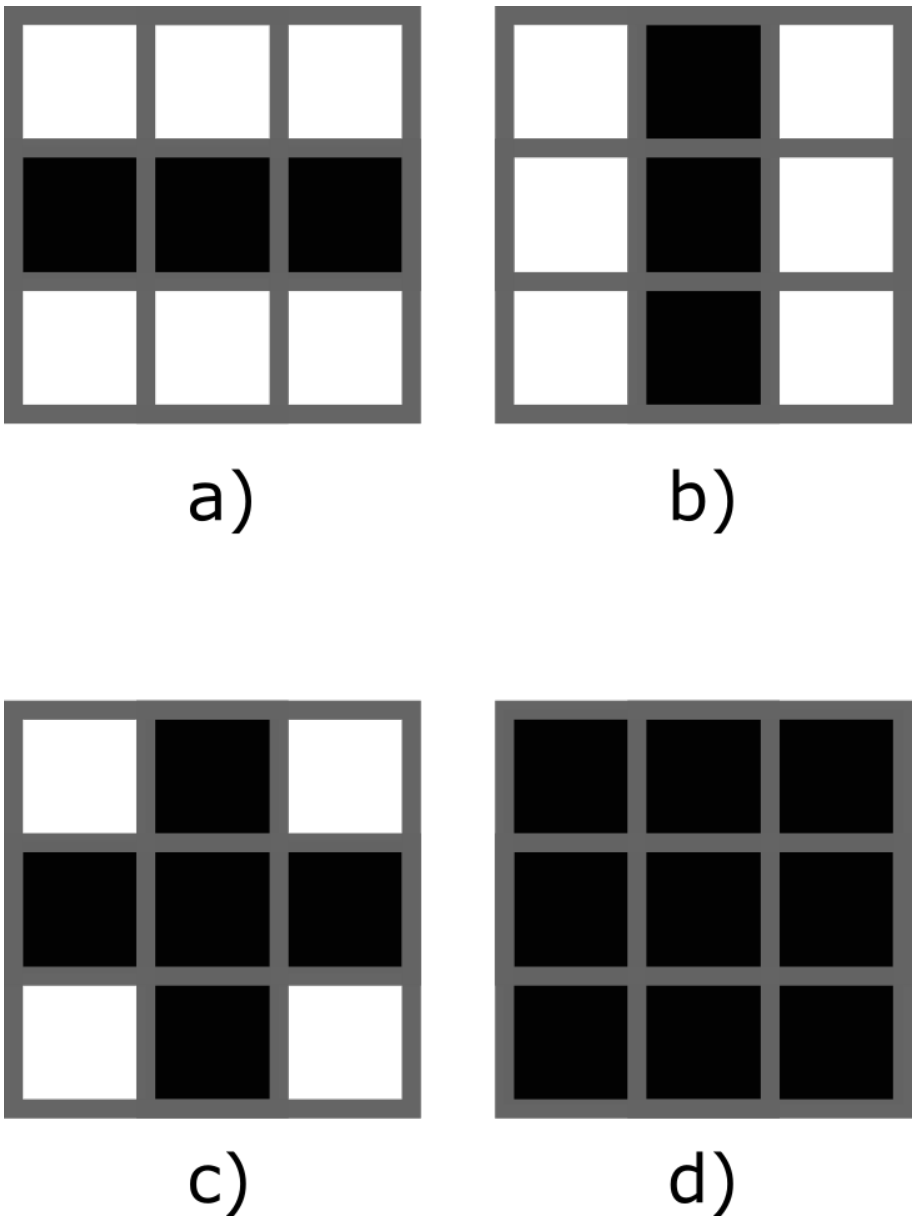
W przedstawionym przykładzie metoda Niblacka daje najlepszy rezultat spośród wszystkich trzech opisanych metod. Algorytmy Sauvoli oraz Niblacka, jako metody lokalne, odznaczają się lepszym dostosowaniem do charakterystyki obrazu. Wobec tego oczekiwanym rezultatem jest większa odporność w przypadku operowania na dokumentach z niejednolitym oświetleniem.

3.3 Morfologia

Morfologia jest narzędziem matematycznym, jedną z najważniejszych i najczęściej używanych operacji podczas przetwarzania obrazów. Może posłużyć m. in. do detekcji obiektów [33]. Stosowanie tej operacji ma na celu zmodyfikować strukturę bądź kształt obiektu. Operacje morfologiczne mogą zostać określone na podstawie teorii zbiorów. W tym celu obraz należy postrzegać jako zbiór punktów. Każdy punkt jest reprezentowany poprzez swoje współrzędne (x, y) w dwuwymiarowej przestrzeni $\mathbb{Z}^2$.

Zazwyczaj operacje morfologiczne wykorzystują maskę, nazywaną elementem strukturalnym. Jest to grupa pikseli, która reprezentuje pewien obiekt, np. linię, diament, kwadrat. Element strukturalny charakteryzuje się punktem centralnym. Jest to punkt dzielący maskę na pik-

sel aktualnie przetwarzany oraz jego sąsiedztwo. Zazwyczaj ów punkt jest środkiem ciężkości obiektu wewnątrz elementu strukturalnego. Przykłady takich elementów przedstawia Rysunek 26, punktem centralnym jest komórka w środkowym wierszu i środkowej kolumnie.



Rysunek 26: Przykładowe elementy strukturalne, punktem centralnym jest komórka w środkowej kolumnie i środkowym rzędzie. Przedstawiają one następujące obiekty: a) pozioma linia; b) pionowa linia; c) krzyż; d) kwadrat

Dwiema podstawowymi operacjami morfologicznymi są dylatacja (6) i erozja (7). Ich definicje zostały przedstawione poniżej:

$$A \oplus B = \max_i [A + B(i)] \tag{6}$$

$$A \ominus B = \min_i [A - B(i)] \tag{7}$$

gdzie:

A - reprezentuje piksele obiektów

B(i) - reprezentuje element strukturalny o punkcie środkowym i

Niniejsza praca zakłada, że piksele obiektu mają kolor czarny, piksele tła biały. Operacje morfologiczne mogą zostać użyte do eliminacji niechcianych artefaktów z obrazu. Można także wykorzystać je do złączenia przerw [33]. W szczególności, operacja erozji może usunąć niewielkie struktury lub wąskie linie. Z kolei operacja dylatacji pozwala zwiększyć obiekt. Może zostać użyta w celu wyeliminowania małych przerw wewnątrz obiektu. Pozwala także złączyć struktury, które zostały przerwane wskutek zastosowania wcześniejszych operacji. Głównym problemem dylatacji jest fakt, że zwiększa ona powierzchnię obiektu, z kolei erozja zmniejsza ją. Takie zachowanie jest często niepożądane. Z tego powodu częstą praktyką jest łączenie obu operacji. Zastosowanie erozji, a następnie dylatacji nazwane jest otwarciem morfologicznym, z kolei zastosowanie dylatacji, a później erozji nazwane jest zamknięciem morfologicznym. Takie operacje eliminują główną wadę dylatacji i erozji - ich zastosowanie nie zmienia znacząco powierzchni obiektu.

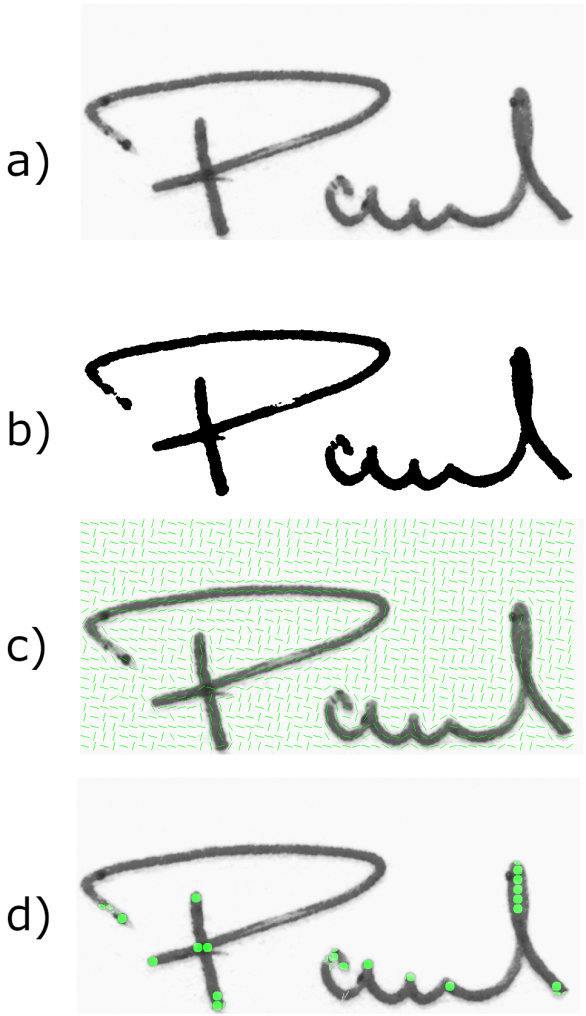
Niemniej jednak, żadna z przedstawionych powyżej metod nie jest odporna zarówno na problem przerwania linii pierwotnie połączonej jak i problem zakrywania luk, które są ważną częścią obiektu. Częstą przyczyną tych błędów jest jednakowy kształt i rozmiar elementu strukturalnego wykorzystywanego przez operację morfologiczną. Niniejsza rozprawa przedstawia sposób na dostosowanie elementu strukturalnego do cech obiektu z wykorzystaniem pola kierunkowego.

3.4 Pole kierunkowe

Algorytm opisany w niniejszej rozprawie wykorzystuje pole kierunkowe [100] w celu przewidzenia kierunku aktualnie analizowanej linii. Mając taką dodatkową informację, algorytm jest w stanie przewidzieć dalszy oczekiwany kierunek linii w miejscu, w którym została ona niepoprawnie usunięta podczas binaryzacji. Oczekiwanym rezultatem analizy pola kierunkowego jest przede wszystkim naprawa zerwanych połączeń. Takie problemy były wykazywane przez binaryzację Sauvoli i Niblacka w rozdziale 3.2. Poniżej zostały przedstawione trzy sposoby wyznaczenia pola kierunkowego.

3.4.1 Podejście oparte na gradiencie z miarą koherencji

W tej metodzie, dla każdego piksela obrazu wejściowego (x, y) analizowane jest sąsiedztwo. Na podstawie tego obliczana jest wartość $\theta(x, y)$ - wartość pola kierunkowego w danym punkcie (informacja o kierunku linii w tym punkcie). Przykładowy obraz z wizualizacją pola kierunkowego został przedstawiony na Rysunku 27 (c).



Rysunek 27: Kolejne procesy przetwarzania przykładowego podpisu: a) Obraz w odcieniach szarości; b) Obraz po binaryzacji; c) Pole kierunkowe wyznaczone metodą gradientową dla każdego regionu obrazu; d) Obraz z zaznaczonymi na zielono regionami o niskiej wartości miary koherencji

Obraz został podzielony na niewielkie segmenty, każdy z segmentów zawiera czerwoną linię. Kierunek owej linii reprezentuje kierunek dominujący wewnątrz tego segmentu. Można zaobserwować, że segmenty będące częścią pisma reprezentują kierunek tożsamy z kierunkiem napisanej linii w danym segmencie. Segmenty zawierające jedynie piksele tła (niebędące czę-

ścią linii podpisu) należy uznać za szum i mogą zostać zignorowane.

Istnieje wiele metod używanych do wyznaczenia pola kierunkowego. Najczęściej wykorzystywane metody bazują na wyznaczaniu gradientu [57]. Posiadając obraz I w odcieniach szarości, aby obliczyć gradient G_x i G_y wzdłuż osi odpowiednio: x i y, należy użyć następującego wzoru:

$$\begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \tag{8}$$

Istnieje wiele algorytmów, które pozwalają obliczyć gradient. Jednym z rozwiązań jest splot obrazu z maską Prewitta lub Sobela. W niniejszej pracy dodatkowo został wykorzystany filtr Gaussa wzdłuż kierunków osi x i y, który dodatkowo wygładza obraz.

Gradient obliczony dla każdego piksela obrazu jest czuły na niewielkie lokalne zmiany, które mogą być spowodowane m. in. przez szum. Aby zmniejszyć wrażliwość na te zmiany, należy uśrednić wartość gradientu wewnątrz okna o stałym rozmiarze, jak zostało to przedstawione w [38].

Podczas obliczania wartości średniej, gradienty o kierunkach przeciwnych powinny zostać dodane zamiast wygaszania. Wynika to z faktu, że reprezentują one ten sam kierunek linii. Kolejną ważną obserwacją są gradienty o małej wielkości. Mogą one zostać wyeliminowane, gdyż jest to szum nie reprezentujący regionu z krawędzią.

Jednym ze sposobów obliczenia średniej wartości jest zapis wektora gradientu jako liczby zespolonej. Następnie wystarczy obliczyć sumę kwadratów gradientów G_W dla każdej pozycji (x, y) w oknie W obrazu wejściowego, o środku w (x, y):

$$G_W(x,y) = \sum_{(x',y') \in W} (G_x(x',y') + jG_y(x',y'))^2 \tag{9}$$

W celu zmniejszenia wahania się zmian kierunków gradientu, może zostać zastosowany filtr dolnoprzepustowy. W tym celu należy w pierwszej kolejności znormalizować wektory G_W do długości jednostkowej, a następnie zastosować filtr Gaussa, jak pokazuje poniższe równanie:

$$G_S(x,y) = \frac{G_W(x,y)}{G_W(x,y) \otimes K(x,y)} \tag{10}$$

Końcowy kierunek linii w punkcie (x, y) to argument liczby zespolonej G_S:

$$\theta(x,y) = \frac{1}{2} arg(G_S(x,y)) + \frac{\pi}{2} \tag{11}$$

Z racji sumowania kwadratów w równaniu (9), kąt musi zostać podzielony przez 2 i przesunięty o $\frac{\pi}{2}$ (gradient prostopadły do linii). Zaproponowana metoda wykorzystuje $\theta(x, y)$, w celu lokalnego dostosowania liniowego elementu strukturalnego.

W regionach, w których krawędź nie jest określona lub sąsiedztwo zawiera wiele krawędzi w różnych kierunkach (jak np. przecięcia czy okrągłe kształty), wartość obliczona przez pole kierunkowe nie będzie poprawna. Aby ocenić rzetelność pola kierunkowego w danym regionie, wykorzystany zostanie współczynnik koherencji kierunków w sąsiedztwie, określony wzorem:

$$r(x,y) = \frac{|\sum_{(x,y)\in W} G_S(x,y)|}{\sum_{(x,y)\in W} |G_S(x,y)|} \tag{12}$$

Wielkość tej miary przyjmuje wartości w przedziale od 0 do 1, gdzie 1 oznacza idealną koherencję, natomiast 0 wskazuje na brak koherencji.

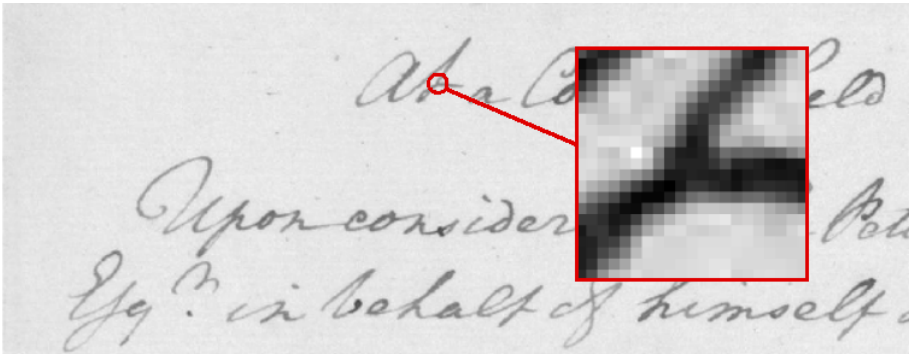
Rysunek 27 (d) ilustruje rezultat zwrócony przez ten współczynnik na zdjęciu przykładowego podpisu. Miejsca oznaczone na zielono reprezentują regiony obrazu o niskim poziomie koherencji. W prezentowanym algorytmie, wartość $r(x,y)$ porównana jest z wybranym heurystycznie progiem. Regiony poniżej tego progu zostały oznaczone na rysunku. Jeśli wartość okaże się większa, algorytm stosuje w pozycji (x,y) liniowy element strukturalny. Jego kierunek jest określany na podstawie wartości funkcji $\theta(x,y)$. W przeciwnym przypadku, kierunek wyznaczony przez $\theta(x,y)$ uznawany jest za nierzetelny i zastosowany zostaje element strukturalny o kształcie diamentu.

Zastosowanie podejścia gradientowego potrafi z powodzeniem zidentyfikować kierunki linii w miejscu, w którym są one jednoznaczne. Ograniczeniem algorytmu może być grubość linii. W przypadku, gdy struktury liniowe są zbyt grube i mieszczą wiele regionów, kierunek linii będzie zwracał rzetelną informację jedynie na krawędziach linii i pikseli tła. W przypadku analizy pisma nie powinno być to problemem, jednak uwzględniając uniwersalne przeznaczenie algorytmu, warto mieć to na uwadze. Rozwiązaniem takiego problemu jest dostosowanie rozmiaru regionu do grubości struktury. Istotnym rozszerzeniem podejścia gradientowego do analizy kierunku jest miara koherencji. Pozwala skutecznie zweryfikować poprawność tego kierunku. Niestety, w przypadku braku dominującej orientacji, algorytm nie próbuje opisać bardziej skomplikowanej struktury. Zamiast tego stosowany jest symetryczny element strukturalny. Jak pokazują wcześniejsze badania (rozdział 3.3), takie podejście może udoskonalić końcowy rezultat, lecz równocześnie może wiązać się z utratą części informacji w newralgicznych regionach.

3.4.2 Podejście oparte na gradiencie z miarą koherencji i histogramem kołowym

Główną wadą metody opartej na gradiencie jest brak dostosowania się algorytmu do regionów obrazu, w których znajdują się co najmniej dwa dominujące kierunki. Przykład takiej sytuacji

został przedstawiony na Rysunku 28. W przypadku takich obszarów metoda opisana w roz-



Rysunek 28: Przykład regionu z niejednoznacznym kierunkiem (z niską wartością miary koherencji)

dziale 3.4.1 nie jest w stanie poprawnie zinterpretować przetwarzanego regionu, stąd stosuje element strukturalny w kształcie diamentu, niezależnie od układu linii na przetwarzanym fragmencie obrazu. Jednak bazując na dostępnych informacjach, istnieje metoda, która identyfikuje wszystkie kierunki w regionie o niskim współczynniku koherencji.

Opisany problem może zostać rozwiązany z wykorzystaniem histogramu kołowego. Jest to często stosowane rozwiązanie problemów bazujących na obrazach zawierających zaokrąglone obiekty [64]. Kluczowym warunkiem do poprawnego działania tej metody jest wyznaczenie punktu przecięcia linii.

Aby wyznaczyć ów punkt, stosowana jest technika przesuwania okna o stałym rozmiarze [15]. Dla każdego okna następuje sprawdzenie, czy punkt środkowy jest punktem przecięcia się linii. Punkt przecięcia zdefiniowany jest jako punkt, który łączy co najmniej trzy linie. Sposób, w jaki następuje identyfikacja linii z punktu centralnego jest pokazany poniżej.

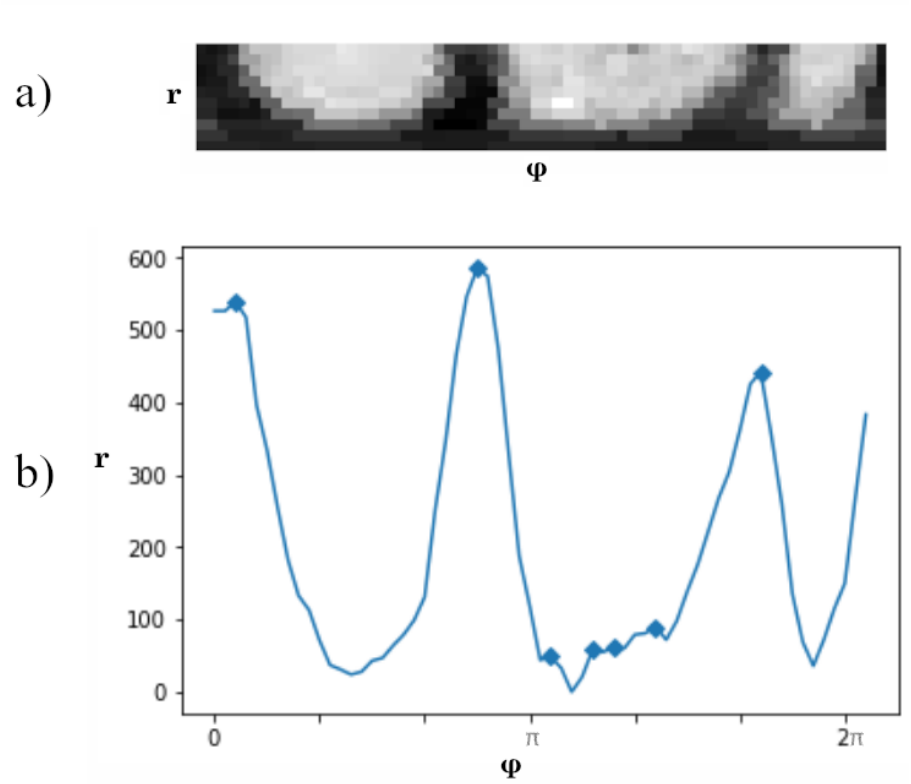
Kolejny krok polega na analizie każdej linii wychodzącej z punktu centralnego. W tym przypadku linia będzie reprezentowana jako para: (φ, r), gdzie $\varphi \in [0, 2\pi]$ jest kątem między linią poziomą wychodzącą z punktu środkowego, r jest długością promienia. Szczegółowe wartości parametrów użytych w eksperymentach zostały przedstawione w rozdziale 5.1.

Para (φ, r) powstaje wskutek zamiany zmiennych x, y ze zmiennych kartezjańskich na biegunowe. Nowe zmienne x_n, y_n powstają poprzez zastosowanie przekształcenia

$$\begin{cases} x_n = rcos\varphi + x_c \\ y_n = rsin\varphi + y_c \end{cases} \tag{13}$$

gdzie (x_c, y_c) jest środkiem aktualnie przetwarzanego regionu, we współrzędnych kartezjańskich.

Rezultaty tej transformacji przedstawione zostały na Rysunku 29 (a).



Rysunek 29: Ilustracja przekształcenia analizowanego regionu obrazu: a) Przekształcenie obrazu ze współrzędnych kartezjańskich na biegunowe; b) Histogram nowo powstałego obrazu

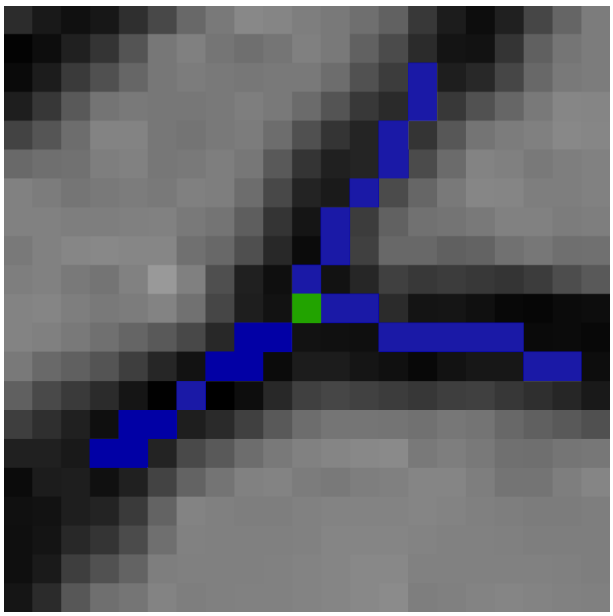
Ostatnim krokiem jest konstrukcja histogramu $h(\phi)$. Dla każdego kąta ϕ i dla określonego r, funkcja h oblicza sumę jasności wszystkich pikseli, które tworzą linię zaczynającą się w punkcie centralnym regionu C i mającą długość równą r. Aby algorytm działał poprawnie, konieczne jest założenie, że jasność obiektu jest mniejsza od jasności pikseli tła. Przykładowe rezultaty zastosowania histogramu kołowego przedstawione są na Rysunku 29 (b). Aby znaleźć lokalne maksimum, należy poddać analizie każdy punkt histogramu oraz jego sąsiedztwa. Warunkiem wystarczającym do znalezienia maksimum lokalnego w punkcie x_0 jest wykazanie, że dla parametru $\delta > 0$, jeden z poniższych warunków jest spełniony:

$$f(x) \texttt{ jest rosnąca dla } x \in (x_0 - \delta, x_0) \tag{14}$$

$$f(x) \texttt{ jest malejąca dla } x \in (x_0, x_0 + \delta) \tag{15}$$

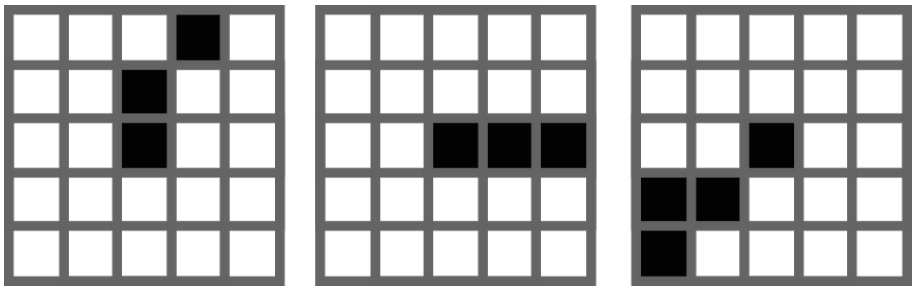
Aby zidentyfikować jedynie poprawne kierunki z histogramu $h(\phi)$, należy określić próg T.

Algorytm używa wartości średniej ze wszystkich wartości na histogramie. Wszystkie maksima powyżej progu T są klasyfikowane jako poprawne. Rysunek 29 (b) pokazuje trzy poprawne maksima, reszta powinna zostać zaklasyfikowana poniżej progu i odrzucona. Rysunek 30 pokazuje linie, które zostały znalezione przez algorytm.



Rysunek 30: Kierunki linii oznaczone na niebiesko zostały zidentyfikowane przez algorytm, szukający istniejących kierunków z punktu oznaczonego kolorem zielonym

Jak pokazuje Rysunek 30, trzy wyróżniające się ekstrema reprezentują wszystkie linie składające się na punkt przecięcia. Na podstawie zdobytej wiedzy, konstruowane są trzy elementy strukturalne, wzmacniające każdą ze znalezionych linii wychodzących z punktu przecięcia. Elementy strukturalne znalezione dla analizowanego przykładu, w rozmiarze 5×5 zostały przedstawione na Rysunku 31. W zależności od rozmiaru elementu strukturalnego, możliwa jest różna



Rysunek 31: Elementy strukturalne utworzone na podstawie analizowanego regionu obrazu

liczba potencjalnych typów połączeń. Z obserwacji wynika, że zastosowana w przykładzie maska 5×5 jest minimalna, by skutecznie naprawiać błędy obrazu.

Zaprezentowany pomysł usprawnia klasyczną metodę opartą na gradiencie, zaprezentowaną w rozdziale poprzednim. Pozwala skutecznie ulepszyć regiony obrazu, w których kierunek linii nie jest jednoznaczny. Ponadto algorytm jest na tyle uniwersalny, że nie ogranicza swojej funkcjonalności do klasycznego przecięcia dwóch linii, reprezentowanego jako punkt z czterema prostymi wychodzącymi z niego. Dzięki możliwości dostosowania rozmiaru elementu strukturalnego i tym samym liczby potencjalnych kombinacji linii, algorytm może dobrze zaadaptować się do różnych zagadnień. Problemem algorytmu wykorzystującego przesuwające się okno jest złożoność obliczeniowa. Jednym ze sposobów zredukowania czasu potrzebnego na przetworzenie pojedynczego zdjęcia jest zmniejszenie jego rozdzielczości. To wiąże się z adekwatnym pomniejszeniem elementu strukturalnego do nowego rozwiązania, co wiąże się z kolejnym przyspieszeniem metody. Inne rozwiązanie zostało zaproponowane w [66]. Autorzy publikacji zauważają, że wykonywanie operacji na oknie często nie jest wymagane w każdej części obrazu. W przypadku obrazów pisma, kluczowe są jedynie elementy obiektu, czyli piksele czarne po procesie binaryzacji. Ograniczenie stosowania metody histogramu kołowego jedynie do tych pikseli oraz do ich najbliższego sąsiedztwa znacząco poprawi wydajność metody, gdyż zazwyczaj piksele tła stanowią znaczną większość pikseli obrazu. Ponadto takie usprawnienie nie wpłynie w żaden sposób na skuteczność metody, gdyż informacja o pozostałych regionach nie jest wykorzystywana w algorytmie.

3.4.3 Podejście oparte na hesjanie

Trzecie rozwiązanie wykorzystuje ideę, którą zaproponował Frangi w swoim algorytmie dla obrazów 2D [31]. W tym przypadku zamiast obliczać pole kierunkowe z wykorzystaniem pierwszych pochodnych, metoda znajduje hesjan i oblicza jego wartości własne λ_1 i λ_2.

Kolejnym krokiem jest weryfikacja, czy badany region jest tłem czy częścią linii. Jest to sprawdzane przy pomocy Tablicy 1.

Algorytm wykorzystuje miarę dla ciemnych krawędzi, którą zaproponował Frangi:

$$\nu_F = \begin{cases} 0 & \texttt{, jeśli } \lambda_2 < 0 \\ \exp(\frac{-R_B^2}{2\beta_2})(1-\exp(\frac{-S^2}{2c^2})) & \texttt{, w przeciwnym przypadku} \end{cases} \tag{16}$$

gdzie $R_B = \frac{\lambda_1}{\lambda_2}$, λ_1 i λ_2 są wartościami własnymi, $\beta = 0,5$ i $S = \sqrt{\lambda_1^2 + \lambda_2^2}$

Ustalanie, czy aktualnie analizowany obszar jest szumem wymaga przeanalizowania znormalizowanej wartości współczynnika ν_F. Jeśli ma on wartość większą niż $0,25$, zakłada się, że

2D		Kierunek
λ_1	λ_2	
S	S	szum, brak ustalonego kierunku
N	W-	struktura cylindryczna (jasna)
N	W+	struktura cylindryczna (ciemna)
W-	W-	struktura plamy (jasna)
W+	W+	struktura plamy (ciemna)

Tablica 1: Możliwe do zidentyfikowania struktury, w zależności od wartości własnych λ_k (W=wysoka, L=niska, N=szum, zwykle niska wartość; $+/-$ znak wartości własnych). Przy tym zakładamy, że: $|\lambda_1| \leq |\lambda_2|$

jest to część linii. Wówczas kierunek linii jest taki sam jak wektor własny mniejszej wartości własnej. Schemat działania algorytmu można więc przedstawić następująco:

1. Znajduje mniejszą wartość własną
2. Znajduje wektor własny odpowiadający tej wartości własnej
3. Wyznacza pionowy albo poziomy element strukturalny, zależnie od kierunku wektora własnego (analogicznie do podejścia opartego na gradiencie)

Metoda obliczania kierunku na podstawie gradientu jest znacznym usprawnieniem w stosunku do metody opartej na gradiencie. Dzięki zastosowaniu drugiej pochodnej, algorytm zdobywa więcej informacji na temat każdego z analizowanych regionów. Dzięki temu nie potrzebuje wykorzystywać dodatkowych narzędzi, takich jak miara koherencji czy histogram kołowy. Dzięki temu taki algorytm jest także szybszy w porównaniu do metody gradientowej z ulepszeniami. Ponadto w [31] zwrócono uwagę na to, że stosowanie takiej metody jest odporne na niewielkie zakłócenia powodowane przez szum.

3.5 Test 4-sąsiedztwa

Algorytm jest stosowany w celu usunięcia małych artefaktów, większość z nich ma rozmiar jednego piksela. W związku z tym, mogą one zostać wyeliminowane poprzez zastosowanie następującej metody: kolor piksela jest zmieniany na kolor sąsiadów, jeśli jego sąsiedzi nad nim, pod nim, po lewej i prawej stronie mają ten sam kolor.

Jest to metoda bazująca na innych, analogicznych rozwiązaniach spotykanych w literaturze [18, 36]. Z racji swojej specyfiki, metoda nie powinna wpływać na jakość reprezentacji poszczególnych glifów. Ma ona za zadanie skupiać się na powierzchni tła, identyfikować potencjalne fragmenty, będące zbyt małe, by zostać zaklasyfikowane jako obiekt. Usunięcie takich pikseli poprawia czytelność całego obrazu. To sprawia, że może być szczególnie przydatna w sytuacjach, gdy efektem końcowym całego procesu przetwarzania obrazu nie jest klasyfikacja elementów znajdujących się na nim, lecz właśnie jak najbardziej czytelny obraz, który byłby przydatny do analizy przez specjalistę.

3.6 Zwiększenie liczby próbek

W celu zwiększenia liczby możliwych do analizy kierunków, algorytm zwiększa rozmiar wejściowego obrazu. Oryginalny obraz w odcieniach szarości jest powiększany do 200% z wykorzystaniem interpolacji kubicznej [47]. Po tym procesie, szerokość oraz wysokość oryginalnego zdjęcia mają dwukrotnie większą długość. Analogicznie, rozmiar elementu strukturalnego rośnie dwukrotnie. W zależności od rozmiaru początkowego, liczba możliwych do utworzenia liniowych elementów strukturalnych wzrośnie. Przykładowo, dla początkowego rozmiaru 3×3, istnieją dokładnie cztery kombinacje możliwych do uzyskania masek: pionowa, pozioma i dwie ukośne. Wraz ze wzrostem rozmiaru maski, wzrasta liczba kombinacji. Wybór konkretnej kombinacji zależy, zgodnie z pierwotnym założeniem, od kąta linii w aktualnie przetwarzanym fragmencie obrazu.

Ostatnim krokiem algorytmu jest redukcja rozmiaru obrazu do pierwotnych rozmiarów. Każda kolejna czwórka pikseli obrazu jest mapowana na jeden piksel. Pierwszym krokiem jest podział rozszerzonego obrazu na kwadraty S o rozmiarze 2×2 pikseli. Nowy piksel p_n jest obliczany jako minimum ze wszystkich wartości wewnątrz kwadratu S_p:

$$\forall_{p_n} p_n = min(\forall_{p \in S_p} p) \tag{17}$$

Przedstawiona operacja zwiększania liczby próbek może pełnić wiele funkcji. Pierwszą z nich jest normalizacja bazy obrazów do wspólnego wymiaru. Jest to często wymagany krok, szczególnie w sytuacji, gdy różnice między rozdzielczościami są duże. Ten krok ułatwia późniejsze dostosowanie parametrów algorytmu do problemu. Ponadto zabieg powiększania rozdzielczości zdjęcia może skutkować polepszeniem działania całego algorytmu przetwarzania obrazów. Zwiększona rozdzielczość zdjęcia wiąże się ze zwiększoną rozdzielczością wykorzy-

stywanych elementów strukturalnych, co daje możliwość reprezentacji większej liczby linii. Z drugiej strony sam proces zwiększenia próbkowania jest czasochłonny [56]. Wykorzystanie tej metody wraz z algorytmem opartym na gradiencie i histogramie kołowym może znacząco spowolnić proces przetwarzania obrazu. Warto zwrócić na to uwagę, w szczególności dla dużych baz obrazów.

4 Weryfikacja przedstawionych technik

4.1 Miara dokładności (Acc)

Miara dokładności (accuracy) pozwala porównać obraz testowany I_t z obrazem zawierającym rezultat wzorcowy I_w. Jest ona obliczana w następujący sposób:

$$S(I_t, I_w) = \frac{\sum_{(x,y)\in I_w} |sgn(I_w(x,y) - I_t(x,y))|}{\sum_{(x,y)\in I_w} 1} * 100\% \tag{18}$$

Zasada działania zaprezentowanej miary może być postrzegana jako klasyfikacja każdego piksela do jednego z dwóch klas: obiektu albo tła. Takie spojrzenie będzie wykorzystane podczas używania narzędzi statystycznych.

4.2 Ulepszona miara dokładności (Acc2)

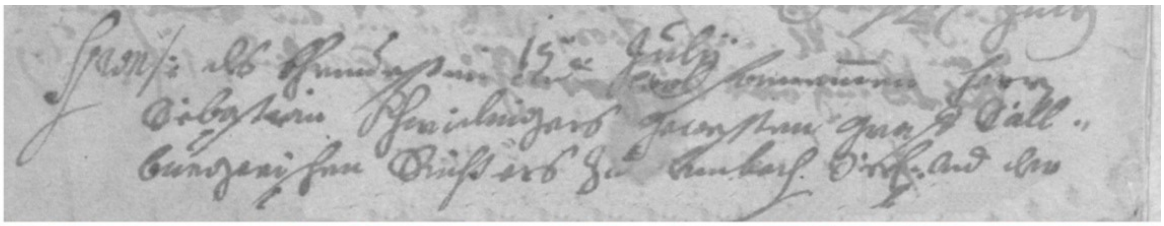
Metoda testowania oparta na weryfikacji każdego piksela z obrazu ma znaczącą wadę, pomimo precyzyjnego porównania podobieństwa obrazu testowanego do wzorcowego. Większość pikseli znajdujących się na obrazie zawierającym podpis zazwyczaj należy do tła. Tym samym ich klasyfikacja nie jest wymagająca. To sprawia, że nawet kiepskie algorytmy są w stanie osiągnąć dobry wynik przy użyciu miary Acc, a różnice między poszczególnymi algorytmami mogą być mniejsze i przez to złudnie sugerujące o podobieństwie w skuteczności działania poszczególnych algorytmów. Naturalna wydaje się potrzeba utworzenia miary, która będzie skupiać się na pikselach najważniejszych z punktu widzenia segmentacji pisma, czyli tych należących do podpisu oraz jego najbliższego sąsiedztwa.

W celu identyfikacji tych pikseli, zaproponowana miara Acc2 używa obraz, na którym została wykonana binaryzacja globalna, a następnie dylatacja. Czarne piksele powstałe w ten sposób tworzą maskę, która zawiera część obrazu z pikselami, które będą testowane w sposób analogiczny do miary Acc.

Dla przykładowego obrazu zaprezentowanego na Rysunku 32. kolorem czerwonym został oznaczony obszar, który będzie uwzględniony przez miarę Acc2. Pozwoliło to zmniejszyć liczbę przetwarzanych pikseli z 595 188 do 305 428. Jest to wynik mniejszy o niemal połowę.

Metoda daje rezultat, który jest bardziej wrażliwy na niedokładności działania algorytmów, takie jak brak ciągłości linii czy niepoprawna grubość linii pisma. Kolejną zaletą przedstawionej miary jest fakt, że uwzględnia ona także elementy bez tekstu, które są trudne do procesowania. Najczęściej występujące artefakty tego typu to: plamy, zagięcia papieru czy cienie.

a)



b)

Rysunek 32: Obszar uwzględniany przy liczeniu miar dokładności: a) obraz wejściowy, w całości uwzględniany przez Acc; b) obszar zaznaczony na czerwono uwzględniany przez miarę Acc2

4.3 Test Friedmana

Test Friedmana jest testem statystycznym, który pozwala zweryfikować, czy różnice w dokładnościach poszczególnych algorytmów są statystycznie istotne [21].

Algorytm porównuje k algorytmów, z których każdy został poddany N testom. Ocena działania algorytmu polega na nadaniu oceny każdemu z algorytmów: najlepszy ze wszystkich dostaje ocenę 1, drugi otrzymuje 2 itd. Ostateczna ocena j-tego algorytmu jest określona następująco:

$$R_j = \frac{1}{N} \sum_i r_i^j \tag{19}$$

Ostateczna wartość statystyki Friedmana jest obliczona przy pomocy następującego wzoru [26]:

$$\chi_F^2 = \frac{12n}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{20}$$

Obliczona wartość powinna być porównana z wartością krytyczną odczytaną z tablicy rozkładu χ^2. Poziom prawdopodobieństwa przyjęty do obliczeń to 5%. Liczba stopni swobody zależna jest od liczby analizowanych algorytmów i wynosi $k - 1$ (gdzie k jest liczbą testowanych algorytmów).

Jeśli obliczona wartość jest mniejsza od wartości krytycznej, hipoteza zerowa, mówiąca, że wszystkie algorytmy są sobie równoważne nie może zostać odrzucona.

4.4 Test Nemenyi

Nemenyi jest testem post hoc, który może zostać przeprowadzony, gdy hipoteza zerowa została odrzucona. Metoda pozwala zidentyfikować, które pary algorytmów są sobie równoważne [21].

Sposób działania testu opiera się na porównaniu każdej kombinacji dwóch algorytmów spośród wszystkich k. Dwa algorytmy różnią się w sposób istotny statystycznie, jeśli różnica między ich ocenami jest większa lub równa wartości różnicy krytycznej (CD):

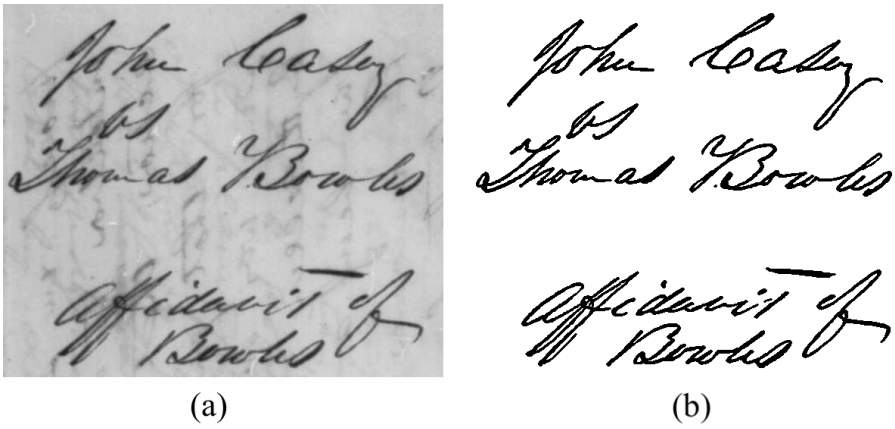
$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{21}$$

gdzie $\alpha = 0,05$.

5 Testy i wyniki

Testy zostały przeprowadzone na trzech bazach: ICFHR (2012) [2], DIBCO (2009-2016) [1] i PBOK [5]. ICFHR zawiera zdjęcia podpisów odręcznych. Baza ta została użyta w celu wstępnej oceny rezultatów algorytmu, zademonstrowania sposobu działania algorytmów, porównania z innymi metodami i zidentyfikowania jego zalet i wad. Baza została użyta, aby ocenić, w jaki sposób algorytm radzi sobie trzymaniem ciągłości linii podpisu czy z szumem.

Z baz DIBCO [1] zostało wybranych czterdzieści osiem zdjęć, po siedem z każdego roku (z wyjątkiem roku 2015, w którym nie dodano nowych zdjęć). Każdy obraz z tej bazy zawiera obraz wzorcowy. Dzięki temu możliwa jest analityczna ocena jakości algorytmu, z wykorzystaniem miar Acc i Acc2. Przykładowa para obrazów (przed procesowaniem i oczekiwany wzorcowy rezultat) została przedstawiona na Rysunku 33.



(a) (b)

Rysunek 33: Przykładowa para obrazów z bazy DIBCO: a) obraz w odcieniach szarości; b) oczekiwany obraz po przetworzeniu

Ostatnią wykorzystywaną bazą obrazów jest PBOK. Baza zawiera obrazy z tekstem w językach niekorzystających z alfabetu łacińskiego. Wybrano z niej sto zdjęć z tekstem w języku perskim. Podobnie do bazy DIBCO, baza PBOK także zawiera obrazy wzorcowe, co umożliwia zastosowanie miar analitycznych do oceny jakości algorytmu.

5.1 Parametry algorytmu

Przedstawiony algorytm jest na tyle uniwersalny, że może zostać zastosowany do szerokiego spektrum problemów segmentacji obiektów z obrazu. W zależności od typu problemu, do poprawnego działania należy dostosować parametry algorytmu. Instrukcje przedstawione w ni-

niejszym rozdziale zostały dostosowane do rozpoznawania pisma ręcznego, na podstawie trzech dostępnych baz. Dla innych problemów (jak np. obrazy medyczne) zaproponowane wartości nie muszą dawać najlepszych rezultatów.

Rozmiar elementu strukturalnego zależy od krzywizny litery. Najlepsze rezultaty można uzyskać stosując maskę o szerokości boku stanowiącej 25% wysokości znaku. Aby poprawnie wypełnić nieciągłości linii powstałe wskutek binaryzacji, istotnym założeniem jest rozmiar liniowego elementu strukturalnego (powodującego dylatację), który musi być większy od kwadratowego elementu strukturalnego (powodującego erozję). Druga maska powinna być dwukrotnie mniejsza.

Dla wszystkich przeprowadzonych testów, parametry zaproponowane w metodzie były jednakowe. Zostały wybrane na podstawie serii eksperymentów, biorąc pod uwagę powyższe wnioski.

- Maska 7×7 przy obliczaniu pola kierunkowego
- Maska 15×15 do uśredniania gradientów
- Próg 0.7 dla współczynnika koherencji
- Maska 5×5 dla liniowego elementu strukturalnego
- Maska 5×5 dla elementu strukturalnego w kształcie diamentu
- Maska 3×3 dla kwadratowego elementu strukturalnego używanego do erozji
- Promień 10 podczas konwersji współrzędnych z kartezjańskich na biegunowe podczas obliczania histogramu kołowego
- Krok $\phi = 0.1$ podczas tworzenia histogramu kołowego

5.2 Eksperymenty I (ICFHR)

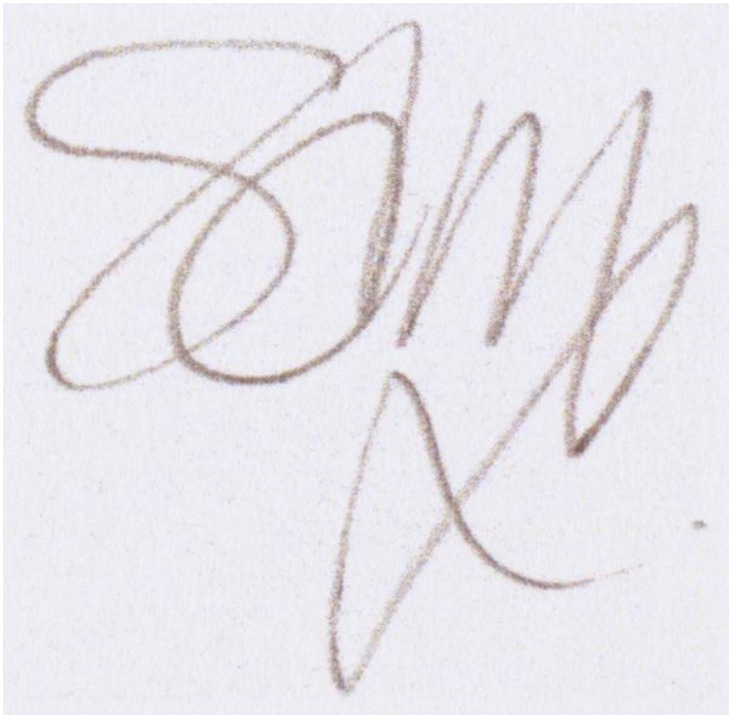
Poniższe testy pokazują rezultaty algorytmu obliczającego pole kierunkowe z wykorzystaniem podejścia opartego na gradiencie, z miarą koherencji, histogramem kołowym i testem 4-sąsiedztwa. Podejście zostało porównane z tradycyjnymi podejściami do preprocessingu:

- dylatacja (kwadrat)
- erozja (kwadrat)

- filtr medianowy (diament)
- zamknięcie morfologiczne (kwadrat)
- otwarcie morfologiczne (kwadrat)

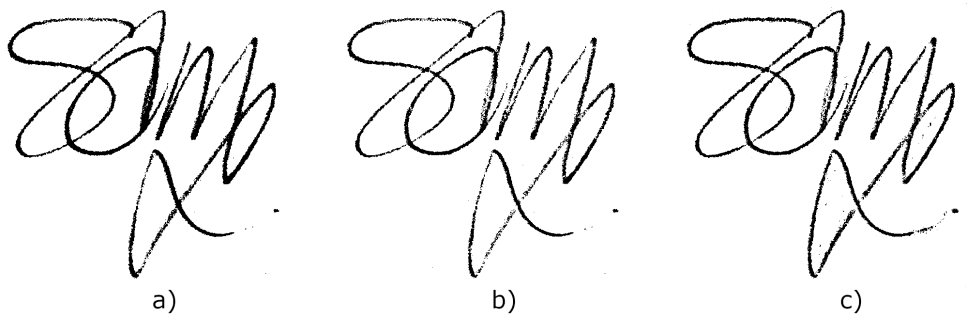
We wszystkich tradycyjnych podejściach przedstawionych powyżej została zastosowana maska o rozmiarze 3×3. Do testów użyto różnych algorytmów binaryzacji: metody Otsu, Sauvoli i Niblacka.

Testy zakładają zestawienie każdego algorytmu postbinaryzacyjnego z każdą metodą binaryzacji. Testom został poddany obraz przedstawiony na Rysunku 34. Charakteryzuje się on mniejszym kontrastem koloru pisma oraz tła w porównaniu do innych baz danych. Co więcej, kolor pisma, a także jego grubość nie jest równomierna, co powinno być dodatkowym wyzwaniem. Chropowata struktura papieru, na którym utworzono prezentowany podpis powinna utrudniać zachowanie spójności na obrazie po binaryzacji. Weryfikacja poprawności odwzorowania nieregularnych kształtów i przecięć linii wewnątrz podpisu będzie kryterium oceny poszczególnych metod postbinaryzacji.



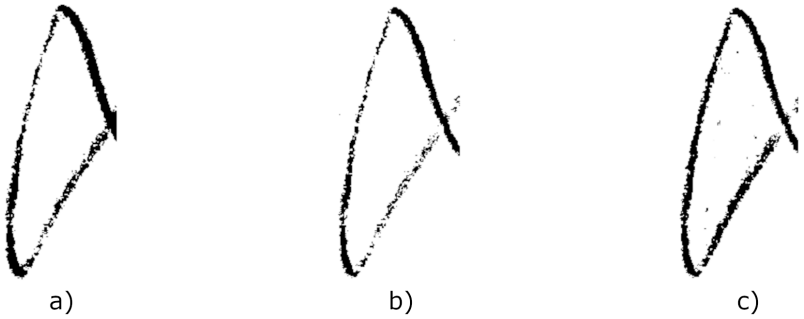
Rysunek 34: Obraz z bazy ICFHR, na którym zostaną zaprezentowane wyniki testów.

W celu dokładniejszej oceny, metody postbinaryzacji zostaną przetestowane na rezultatach trzech różnych algorytmów binaryzacji: Otsu, Sauvoli oraz Niblacka. Efekty działania samej



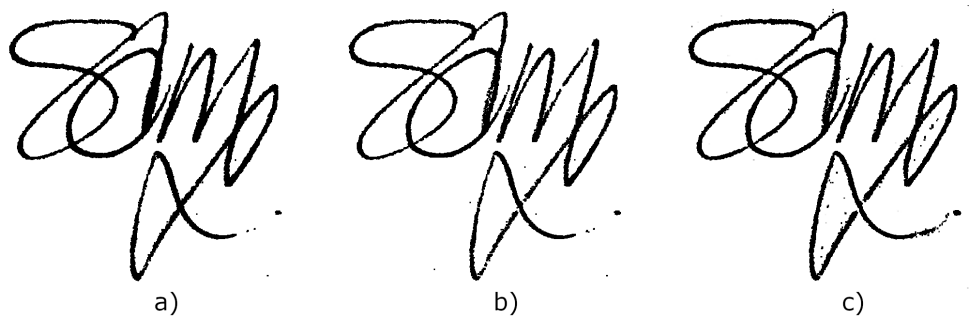
Rysunek 35: Porównanie trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

binaryzacji zostały przedstawione na Rysunku 35. Można zauważyć, że żadna z metod nie daje zadowalających rezultatów, co narzuca potrzebę stosowania metod postbinaryzacji. Najlepsze wyniki w tym przypadku uzyskała metoda Otsu, która okazuje się być najmniej wrażliwa dla niewielkiego kontrastu pikseli obiektu i tła. Metoda Sauvoli zawiera najwięcej przerw i niedoskonałości, które powinny być poprawione przez algorytmy postbinaryzacji. Żadna z metod nie gwarantuje zachowania ciągłości linii. Można zaobserwować w dolnych partiach obrazów, że dla każdego algorytmu binaryzacji linia traci swą ciągłość. Problem został dokładnie zobrazowany na Rysunku 36.

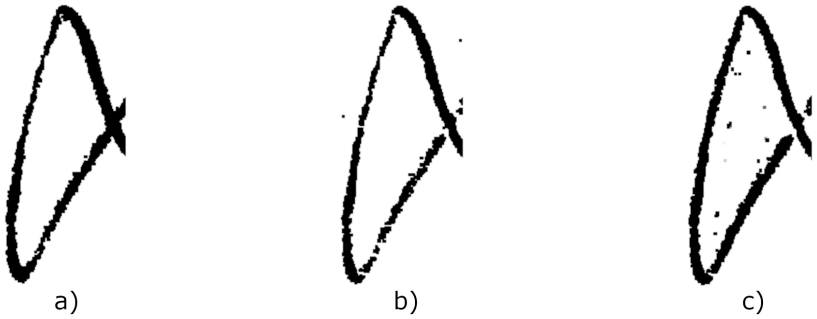


Rysunek 36: Problem ciągłości linii zaprezentowany na trzech metodach binaryzacji: a) Otsu b) Sauvoli c) Niblacka

Sposobem na wyeliminowanie tych nieciągłości jest zastosowanie dylatacji. Poprzez rozpropagowanie pikseli obiektu na sąsiedztwo, przerwy wynikające z chropowatej powierzchni kartki powinny zostać wyeliminowane. Rezultat zastosowania metody dylatacji został przedstawiony na Rysunku 37. Warto zauważyć, że problem nieciągłości linii został znacząco zredukowany. Dokładnie zostało to pokazane na rysunku 38. Jednak zastosowanie dylatacji tworzy dodatkowe błędy na obrazie. Z racji zwiększenia obszaru obiektu, możliwa jest utrata informacji na temat niewielkich pętli bądź mogą zostać złączone równoległe linie biegnące blisko siebie.

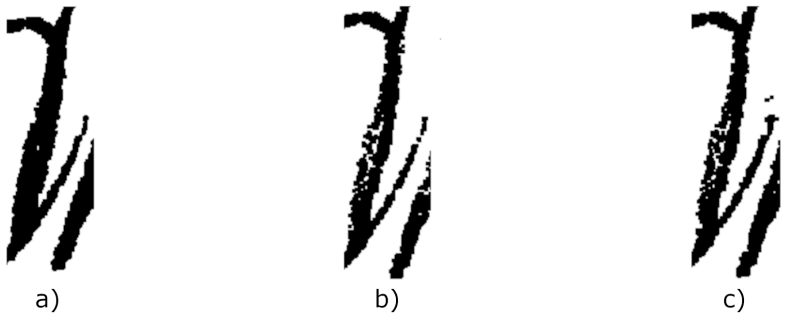


Rysunek 37: Porównanie rezultatów dylatacji dla trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka



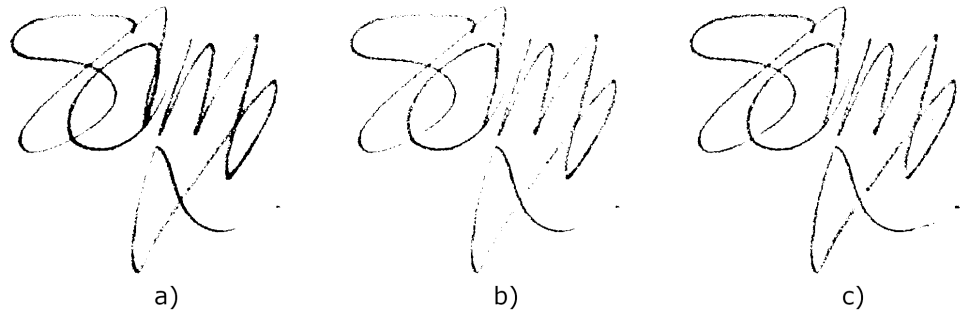
Rysunek 38: Problem ciągłości linii dla dylatacji zastosowanej na trzech metodach binaryzacji: a) Otsu b) Sauvoli c) Niblacka

Taki problem został pokazany na Rysunku 39. Binaryzacja Otsu tworzy najgrubsze linie pisma. Przez to niewielka pętla została wyeliminowana. Jeśli byłaby to cecha szczególna pisma, może to uniemożliwić rozpoznanie autora podpisu.



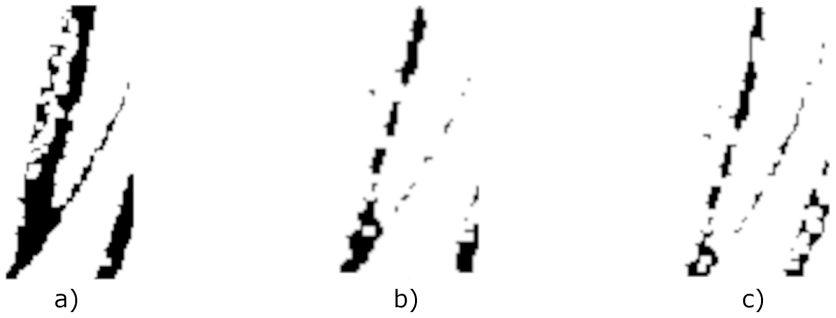
Rysunek 39: Problem niewielkich pętli dla dylatacji zastosowanej na trzech metodach binaryzacji: a) Otsu b) Sauvoli c) Niblacka

Operacją przeciwną do dylatacji jest erozja. Zastosowanie jej pozwala utrzymać informację o wewnętrznych pętlach. Chroni ona także przed niepożądanym złączeniem się równolegle biegnących linii. Rysunek 40 prezentuje efekt jej działania. Należy zwrócić uwagę na dobrze



Rysunek 40: Porównanie rezultatów erozji trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

odwzorowaną pętlę przy metodzie Otsu. Zostało to wyszczególnione na obrazie 41. Jednak

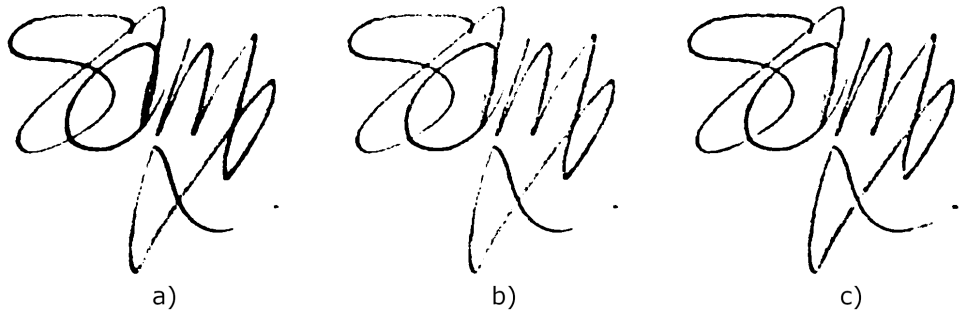


Rysunek 41: Problem niewielkich pętli dla erozji zastosowanej na trzech metodach binaryzacji: a) Otsu b) Sauvoli c) Niblacka

zastosowanie erozji znacząco pogarsza spójność linii i w niektórych regionach obrazu może uniemożliwić identyfikację tekstu, co widać w szczególności na obrazie z zastosowaną binaryzacją Sauvoli. Pewnym rozwiązaniem mogłoby być stosowanie erozji lokalnie, w miejscach, w których istnieje niebezpieczeństwo utraty informacji wskutek dylatacji. Jednak zastosowanie jej na całym obrazie nie daje satysfakcjonujących rezultatów.

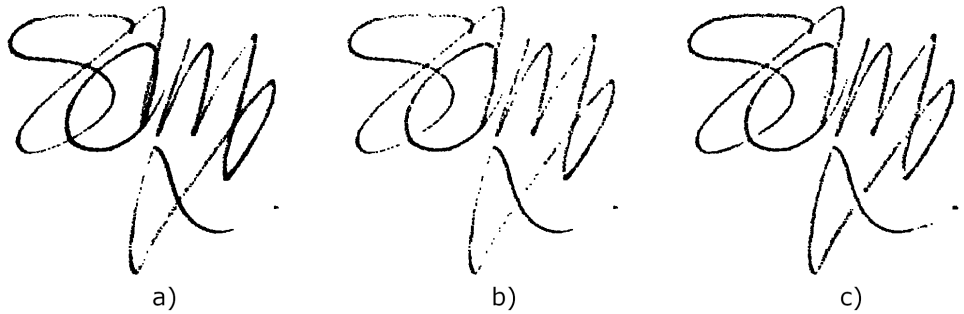
Innym często stosowanym algorytmem preprocessingu jest filtr medianowy. Efekty jego działania prezentuje Rysunek 42. Ów filtr sprawia, że krawędzie linii są mniej ostre. Pismo staje się bardziej czytelne, spójność linii uległa poprawie. Niestety, w porównaniu z dylatacją spójność prezentuje się gorzej, co można zauważyć w dolnych regionach obrazów. Informacja o pętli została utracona, co jest rezultatem gorszym w porównaniu z erozją. Filtr medianowy może być więc traktowany jak pewien kompromis między erozją i dylatacją, gdyż zawiera część zalet obu podejść, lecz niestety także część ich wad.

Podobnych rezultatów można się spodziewać poprzez zastosowanie kombinacji erozji, a na-



Rysunek 42: Porównanie rezultatów filtru medianowego dla trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

stępnie dylatacji (otwarcia morfologicznego) lub dylatacji i następnie erozji (zamknięcia morfologicznego). Rysunek 43 prezentuje wyniki otwarcia. Można zauważyć, że podejście zacho-

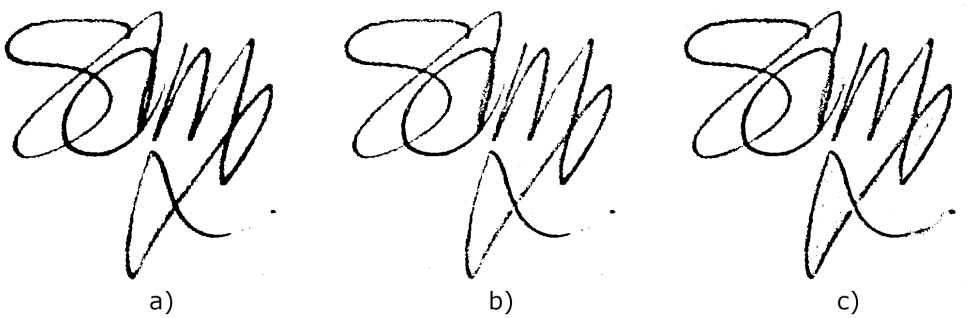


Rysunek 43: Porównanie rezultatów otwarcia morfologicznego dla trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

wuje zalety erozji. W przypadku binaryzacji Otsu, informacja o pętli jest dobrze zachowana. Niestety, dla binaryzacji Sauvoli takie podejście generuje obraz gorszy w porównaniu do samej binaryzacji.

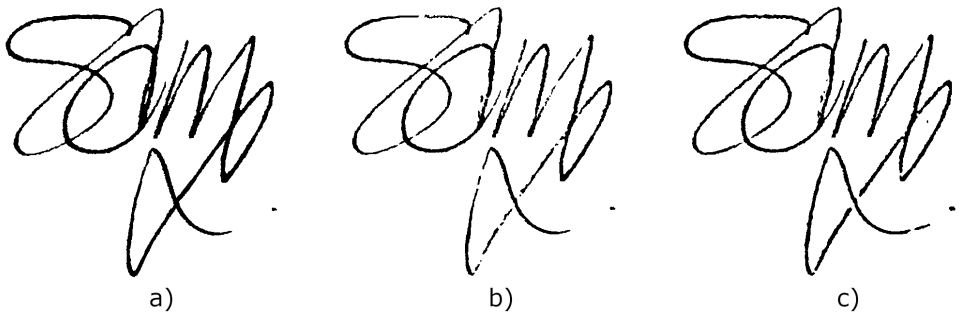
Lepszym podejściu w tym przypadku jest rozpoczęcie filtrem dylatacji, a następnie zastosowanie erozji. Jak pokazuje Rysunek 44, poprawia to efekt każdej z metod binaryzacji. Co więcej zachowana została informacja o pętli na obrazie po binaryzacji Otsu. Jednak wciąż pozostaje problem z ciągłością linii, co widać szczególnie na obrazie po binaryzacji Sauvoli. Ponadto linie są postrzępione, co sprawia, że rezultat odbiega od lepiej prezentującego się filtru medianowego. Innym problemem są niewielkie artefakty, które zostały błędnie zaklasyfikowane jako piksele obiektu. Widać je szczególnie na rezultacie binaryzacji Niblacka. Zamknięcie morfologiczne nie może usunąć takiego szumu, co więcej, uwydatnia go.

Sposobem na uzyskanie najlepszych rezultatów może być zastosowanie algorytmu autorów.



Rysunek 44: Porównanie rezultatów zamknięcia morfologicznego dla trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

Jego rezultaty prezentuje rysunek 45.



Rysunek 45: Porównanie algorytmu autorów dla trzech metod binaryzacji: a) Otsu b) Sauvoli c) Niblacka

Jak widać, prezentuje on najlepsze rezultaty spośród innych, zaprezentowanych algorytmów. W większości przypadków zdołał naprawić uszkodzone linie. Zachował informację o pętli na obrazie po binaryzacji Otsu. Linie wskutek stosowania tego algorytmu są mniej postrzępione. Wizualnie obrazy powstałe wskutek zastosowania autorskiego algorytmu wydają się być najbardziej czytelne spośród innych algorytmów.

Niemniej jednak są przypadki, w których algorytm nie zdołał wyeliminować uszkodzenia linii. Są to przypadki, w których rozmiar uszkodzenia jest większy od rozmiaru stosowanego elementu strukturalnego. W wyniku binaryzacji, zależnie od wybranej metody, możliwe jest powstanie dużych dziur bądź utrata ciągłości linii na znacznej przestrzeni w porównaniu do rozmiaru znaku. Innym możliwym problemem są małe pętle, które zostaną zaklasyfikowane jako elementy o niskim współczynniku koherencji. W rezultacie taki obszar zostanie wypełniony, niezgodnie z oczekiwaniami.

Następna część eksperymentów skupia się na porównaniu skuteczności algorytmu przy za-

stosowaniu różnych kombinacji poszczególnych części algorytmu. W szczególności analizowane jest działanie różnych sposobów obliczenia pola kierunkowego, sprawdzany jest wpływ zwiększania liczby próbek na wyniki. W celu uzyskania obiektywnej oceny, testy przeprowadzone są na bazach udostępniających także obrazy wzorcowe, z wykorzystaniem miary Acc i Acc2.

5.3 Eksperymenty II (DIBCO, PBOK)

Bazy danych DIBCO i PBOK zawierają zdjęcia z tekstem pisanym ręcznie. Z racji dostępu do wzorcowych obrazów po poprawnym preprocessingu, istnieje możliwość oceny skuteczności działania algorytmu z wykorzystaniem miar przedstawionych w rozdziałach 4.1 i 4.2.

Pierwszy test porównuje algorytm wyliczający pole kierunkowe metodą opartą o gradient z miarą koherencji z różnymi, standardowymi technikami preprocessingu. Dla każdej z tych metod (dylatacja, otwarcie i zamknięcie morfologiczne, filtr medianowy) został zastosowany element strukturalny o rozmiarze 5×5.

Prezentowany algorytm używa dylatacji z elementem strukturalnym o różnych rozmiarach. Warto zauważyć, że użycie jedynie dylatacji pogrubia obiekt. Stąd konieczne jest użycie zamknięcia. Z tego powodu ostatnim krokiem algorytmu jest zastosowanie erozji. Użyty w tym celu został element strukturalny o stałym rozmiarze 3×3 pikseli.

Algorytmy zostały przetestowane z różnymi typami binaryzacji. Ostateczne rezultaty przedstawia Tablica 2.

Pierwszy wiersz przedstawia wyniki działania samej binaryzacji. Kolejne wiersze prezentują wyniki dla kolejnych rodzajów preprocessingu, w tym porównywane są różne wersje algorytmu prezentowanego w niniejszej pracy. Warto zwrócić uwagę na znaczące różnice między miarami Acc i Acc2. W przypadku tej pierwszej, operującej na wszystkich pikselach obrazu, dla wszystkich testowanych metod binaryzacji otrzymano skuteczność ponad 80%. Z kolei dla miary Acc2 skuteczność ta jest niższa o nawet 20 punktów procentowych. Istotną obserwacją jest fakt, że miara Acc daje wyższą skuteczność niż Acc2 dla dowolnej binaryzacji i dowolnej metody preprocessingu. To świadczy o tym, że miara Acc2 została skonstruowana poprawnie i skupia analizę poprawności na pikselach trudniejszych do klasyfikacji. Tym samym miara Acc2 lepiej pokazuje różnice w skuteczności między poszczególnymi algorytmami. Następny rozdział pokaże, czy wyniki uzyskane przez tę miarę są istotne statystycznie.

		Otsu		Sauvola		Niblack	
		Acc [%]	Acc2 [%]	Acc [%]	Acc2 [%]	Acc [%]	Acc2 [%]
1	Bez preprocessingu	81,90	66,69	82,36	68,67	81,59	68,50
2	Dylatacja (diament)	83,94	73,47	83,35	68,74	83,45	69,01
3	Dylatacja (kwadrat)	84,48	72,72	83,51	71,95	83,64	72,75
4	Zamknięcie morfologiczne (diament)	83,57	71,59	85,31	74,62	85,72	77,08
5	Otwarcie morfologiczne (diament)	84,13	72,19	84,37	72,08	85,21	72,37
6	Filtr medianowy	84,76	74,21	84,68	72,58	85,56	75,28
7	Metoda autorska (podejście z gradientem, miara koherencji)	89,69	84,47	91,16	87,37	91,33	85,12
8	Metoda autorska (podejście z gradientem, miara koherencji, test 4-sąsiedztwa)	90,06	85,20	91,19	86,12	92,53	87,65
9	Metoda autorska (podejście z gradientem, miara koherencji, histogram kołowy, test 4-sąsiedztwa)	90,34	82,22	91,58	83,85	93,77	89,77
10	Metoda autorska, (podejście z hesjanem, test 4-sąsiedztwa)	90,08	85,16	91,32	83,19	92,57	85,59
11	Metoda autorska (podejście z gradientem, miara koherencji, histogram kołowy, test 4-sąsiedztwa, zwiększenie liczby próbek)	91,80	85,52	92,96	90,5	94,02	91,36

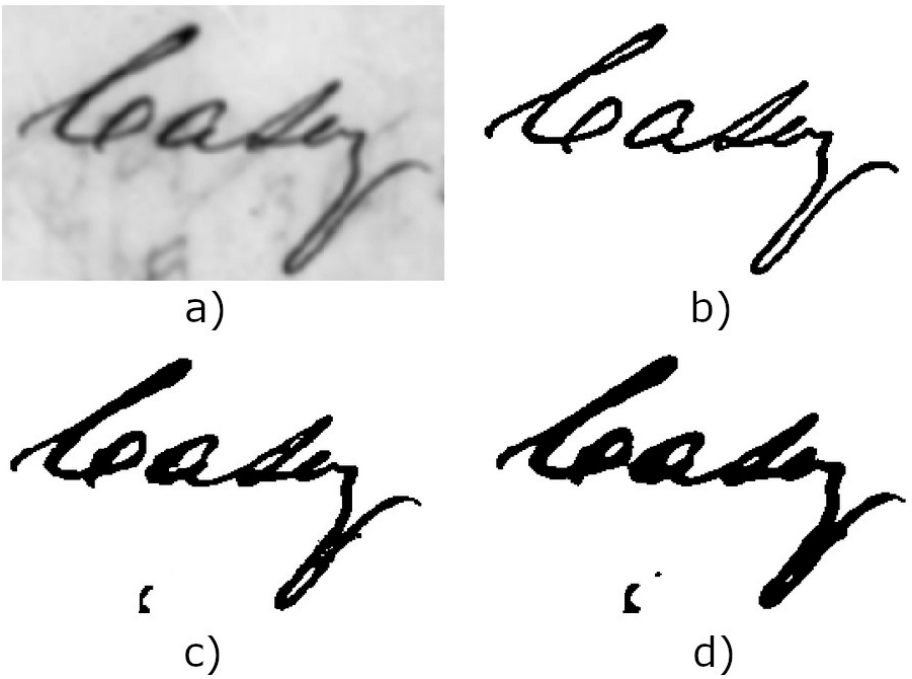
Tablica 2: Miary dokładności (Acc i Acc2) dla wybranych metod binaryzacji i algorytmów preprocessingu, w porównaniu do podejścia autorskiego

5.4 Porównanie z filtrem Frangi'ego

Filtr Frangi'ego [31] jest wykorzystywany do identyfikacji struktur naczyń krwionośnych. Istotnym wymaganiem w przypadku operowania na obrazach medycznych jest odporność na szumy. Filtr Frangi'ego jest jednym z popularniejszych narzędzi, które wykorzystuje się do tego typu

problemów.

Algorytm Frangi'ego jest stosowany na obrazach w odcieniach szarości. W porównaniu zostanie sprawdzona skuteczność działania na obrazach pisma tego filtru z metodą zaproponowaną w niniejszej pracy, wersji z gradientem i histogramem kołowym. Testy zostaną przeprowadzone na bazie danych DIBCO. Zawiera ona obrazy, które mogą być częściowo rozmyte. To pogarsza rezultaty algorytmu binaryzacji, który generuje dla takich zdjęć dużo niechcianego szumu. W przypadku podejścia autorskiego, taki szum powinien być identyfikowany przez test 4-sąsiadów. Filtr Frangi'ego powinien być odporny na szum, wobec tego oczekiwanym rozwiązaniem jest także identyfikacja takich niewielkich artefaktów i eliminowanie ich. Przykładowe porównanie efektów działania obu algorytmów zostało przedstawione na Rysunku 46.



Rysunek 46: Obraz z bazy DIBCO: a) w odcieniach szarości; b) oczekiwane (idealne) rozwiązanie; c) rezultat autorskiego algorytmu; d) rezultat filtra Frangi'ego

Zgodnie z przewidywaniami, filtr Frangi'ego nie daje wystarczająco dobrych rezultatów, w porównania do obrazu wzorcowego. Linie powstałe wskutek zastosowania filtra są pogrubione, co może powodować niewielkie deformacje glifów. Przez to takie rezultaty mogą być trudne do analizy w późniejszych etapach.

Należy zwrócić uwagę na fakt, że wyniki algorytmu autorskiego są zauważalnie lepsze w porównaniu do filtru Frangi'ego dla obrazów pisma. Grubość linii jest bliższa wzorcowej. Ciekawym przypadkiem jest ostatnia litera 'y'. Filtr Frangi'ego spowodował złączenie dwóch rów-

noległych linii. Algorytm autorski w tej sytuacji także pogrubił linię, co spowodowało złączenie się linii w dwóch miejscach. Jednak warto zaznaczyć, że dla danej litery lepsze efekty zostały osiągnięte poprzez zastosowanie algorytmu autorskiego. Analogiczny problem można zauważyć przy pierwszej literze, 'b'. W tej sytuacji oba algorytmy zamknęły pętlę wewnątrz górnej części litery. W przypadku podejścia autorskiego, ten błąd może zostać poprawiony poprzez zastosowanie elementu strukturalnego o większych rozmiarach niż maska 5×5 stosowana podczas testów.

Co ciekawe, żaden z porównywanych algorytmów nie zdołał całkowicie wyeliminować szumu. Świadczy o tym niewielki artefakt poniżej litery 'a' na Rysunku 46 (c) i (d). Świadczy to o tym, że powstały po binaryzacji element był zbyt duży, by mógł zostać uwzględniony przez test 4-sąsiadów w algorytmie autorskim.

5.5 Analiza statystyczna uzyskanych rezultatów

Weryfikacja wyników algorytmów sprawdzająca, czy różnice w rezultatach zwróconych przez poszczególne algorytmy są statystycznie istotne, zostanie przeprowadzona z użyciem testu Friedmana. Można wyodrębnić $k = 11$ grup (algorytmów przedstawionych w Tablicy 2). Zostało przeprowadzonych $n = 148$ testów. Jest to liczba obrazów w bazach DIBCO i PBOK. Zostały one omówione szerzej we wstępie rozdziału 5.

Pierwszym krokiem jest określenie i zweryfikowanie hipotezy zerowej. Należy udowodnić, że jest ona nieprawdziwa. Analiza została przedstawiona w poniższych punktach:

- H_0: Różnice między miarą Acc2 algorytmów 1-11 nie są statystycznie istotne.
  H_1: Różnice między miarą Acc2 algorytmów 1-11 są statystycznie istotne.
- Średnie oceny poszczególnych algorytmów zostały wymienione w Tablicy 3.
- Ostateczna wartość statystyki Friedmana, zgodnie z równaniem (20) wynosi $\chi_F^2 = 1315$.
- Dla założonego poziomu istotności oraz stopni swobody ($d_f = 10$), wartość krytyczna wynosi $18,3070$.
- Jako że $\chi_F^2 > 18,3070$, hipoteza zerowa zostaje odrzucona na rzecz hipotezy alternatywnej.

	R
1	10,68
2	9,29
3	9,18
4	6,63
5	7,84
6	7,39
7	3,76
8	3,67
9	3
10	3,33
11	1,24

Tablica 3: Średnia ocena algorytmów 1-11

Powyższy test udowodnił, że różnice między algorytmami 1-11 są statystycznie istotne. Następnym krokiem jest analiza każdej pary algorytmów. Celem analizy jest weryfikacja, czy różnica między poszczególnymi parami jest istotna statystycznie. W tym celu wykorzystana będzie analiza post hoc przy pomocy testu Nemenyi.

Porównanie między poszczególnymi parami algorytmów zostało przedstawione w Tablicy 4.

Zgodnie z równaniem (21), różnica między algorytmami jest statystycznie istotna, jeśli wartość jest większa od $CD = 1,24$.

Test Nemenyi dowiódł, że wyniki wszystkich wersji prezentowanego algorytmu (7-11) są znacząco lepsze od podejść podstawowych (1-6). Najlepsza wersja (11) okazała się mieć znaczącą przewagę nad pozostałymi algorytmami.

Eksperymenty pokazują, że przedstawiony algorytm poprawia wyniki i jest wyraźnie lepszy od pozostałych testowanych rozwiązań. Warto zauważyć, że stosowanie jedynie binaryzacji daje najgorsze rezultaty. To implikuje, że metody postbinaryzacyjne znacząco poprawiają wyniki binaryzacji, co zostało dowiedzione z użyciem narzędzi statystycznych.

Kolejny test sprawdzał wpływ 4-sąsiedztwa na na wyniki preprocessingu. W tym celu porównano skuteczność metody z uwzględnieniem testu oraz bez niego (siódmy i ósmy wiersz w Tablicy 2). Jedynym celem stosowania tego testu było usunięcie niewielkich artefaktów (po-

	1	2	3	4	5	6	7	8	9	10	11
1	-	1,39	1,5	4,05	2,84	3,28	6,91	7,01	7,68	7,34	9,44
2	-	-	**0,11**	2,66	1,45	1,9	5,53	5,62	6,29	5,96	8,05
3	-	-	-	2,55	1,34	1,78	5,41	5,51	6,18	5,84	7,94
4	-	-	-	-	**1,21**	**0,76**	2,86	2,96	3,63	3,3	5,39
5	-	-	-	-	-	**0,45**	4,07	4,17	4,84	4,51	6,6
6	-	-	-	-	-	-	3,63	3,72	4,39	4,06	6,16
7	-	-	-	-	-	-	-	**0,09**	**0,76**	**0,43**	2,53
8	-	-	-	-	-	-	-	-	**0,67**	**0,34**	2,43
9	-	-	-	-	-	-	-	-	-	**0,33**	1,76
10	-	-	-	-	-	-	-	-	-	-	2,09
11	-	-	-	-	-	-	-	-	-	-	-

Tablica 4: Porównanie statystycznej różnicy między miarą dokładności każdej pary algorytmów 1-11. Wynik dla par, dla których nie ma różnicy istotnej statystycznie, został oznaczony kolorem szarym.

wstałych wskutek niedokładnej binaryzacji). W rzeczywistości, test nieznacznie poprawia wyniki, o $0,5\%$, co nie jest zmianą istotną statystycznie. Taki wynik był prawdopodobny z racji faktu, że obszarem działania tego algorytmu nie są piksele pisma oraz najbliższe otoczenie, lecz głównie obszar tła, na którym powstał szum. Wobec tego skuteczność jego działania mogła zostać nieuwzględniona w mierze Acc2. Z kolei w przypadku miary dokładności Acc, liczba pikseli zmodyfikowanych przez ten test w porównaniu do wszystkich pikseli obrazu jest nieistotnie mała.

Następną przetestowaną metodą był algorytm z miarą koherencji i histogramem kołowym. Jako że dostosowuje się on także do regionów, które algorytm oparty na gradiencie identyfikował jako niejednoznaczne, oczekiwane było polepszenie wyników. W rzeczywistości, jak pokazuje dziewiąty wiersz Tablicy 2, zastosowana modyfikacja zgodnie z oczekiwaniami poprawia skuteczność algorytmu opartego na gradiencie. Niezależnie od wybranej metody binaryzacji, algorytm pozwala w lepszy sposób imitować obraz wzorcowy.

Kolejny test skupiał się na wersji algorytmu opartej na hesjanie. Celem była weryfikacja, czy zastosowanie obliczeń na drugich pochodnych poprawi wyniki w porównaniu do podejścia wykorzystującego pierwsze pochodne (metoda oparta na gradiencie). Średnia wyników

testów tego podejścia została przedstawiona w dziesiątym wierszu Tablicy 2. Wyniki pokazują, że metoda ma podobną skuteczność do podejścia opartego na gradiencie. Największą zaletą oczekiwaną po tym podejściu powinna być lepsza odporność na szum w porównaniu do podejścia bazującego na gradiencie. Niemniej jednak, rezultaty testów pokazują, że nie ma istotnej statystycznie różnicy między tymi dwoma podejściami.

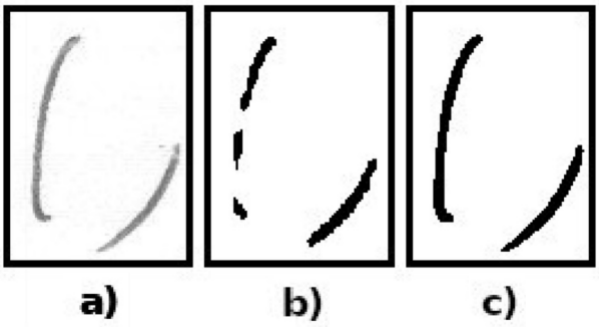
Ostatni test weryfikował zasadność zwiększania liczby próbek. W tym celu wykorzystano metodę opartą na gradiencie i przeprowadzono testy dla każdego typu binaryzacji. Ich wyniki dostępne są w ostatnim wierszu Tablicy 2. Jak można zobaczyć, tak skonstruowana wersja algorytmu pozwala uzyskać najlepsze wyniki spośród wszystkich jedenastu prezentowanych wersji. Jak można zaobserwować, zwiększona liczba analizowanych kierunków (poprzez użycie większej liczby elementów strukturalnych) pozwala lepiej odwzorować poszczególne części linii. Tym samym jest najskuteczniejsza przy naprawianiu błędów powstałych wskutek binaryzacji i pozwala uzyskać rezultaty najbliższe oczekiwanym.

Podczas dostosowywania prezentowanego algorytmu do swojego problemu, należy zwrócić uwagę na szybkość działania metody. Przykładowo, podejście bazujące na hesjanie oraz podejście gradientowe z ulepszeniami są podobne ze statystycznego punktu widzenia. Jednak metoda z wyznaczaniem hesjanu jest szybsza, więc może sprawdzić się lepiej w większości przypadków. W przypadku zwiększania liczby próbek, należy zwrócić uwagę, że metoda może istotnie zwiększyć czas działania algorytmu. Jednocześnie ów etap może nie być konieczny, jeśli pozostałe komponenty algorytmu przetwarzają obraz skutecznie. Warto podkreślić, że prezentowana metoda jest na tyle elastyczna, że pozostawia w gestii programisty decyzję, czy ten skorzysta ze zwiększania liczby próbek kosztem wydłużenia czasu działania, czy zdecyduje się ominąć ten etap.

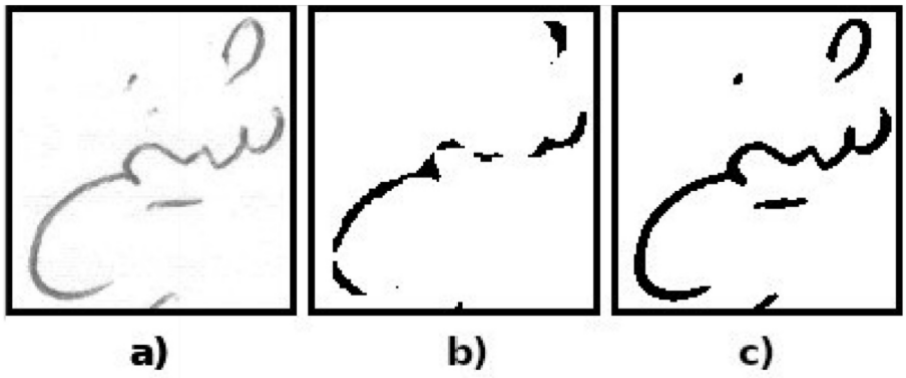
5.6 Eksperymenty z binaryzacją opartą na sieciach konwolucyjnych

Binaryzacja DSN autorstwa Quang Nhat [104] jest algorytmem opartym na głębokich sieciach konwolucyjnych. Był on trenowany na ośmiu publicznych bazach, włączając w to DIBCO. Metoda radzi sobie doskonale z binaryzacją liter alfabetu łacińskiego. Jednak zastosowana do mniej powtarzalnych problemów wykazuje niepoprawne działanie nawet na relatywnie prostych obrazach.

Rysunek 47 oraz Rysunek 48 pokazują przykładowe obrazy glifów arabskich z bazy PBOK [5], dla których nie udało się zachować ciągłości linii. W tym eksperymencie została użyta im-



Rysunek 47: Przykład glifu z bazy PBOK podobnego do alfabetu łacińskiego: a) obraz wejściowy w odcieniach szarości; b) rezultat sieci konwolucyjnych; c) rezultat algorytmu autorskiego



Rysunek 48: Przykład glifu z bazy PBOK różniącego się od alfabetu łacińskiego: a) obraz wejściowy w odcieniach szarości; b) rezultat sieci konwolucyjnych; c) rezultat algorytmu autorskiego

plementacja binaryzacji DSN udostępniona przez samych Autorów w publicznym repozytorium dostępnym w [104].

Zgodnie z wcześniejszymi założeniami, prezentowana metoda może być użyta z dowolnym algorytmem binaryzacji. Jak można zauważyć na Rysunku 47 (c) i Rysunku 48 (c), przedstawiony algorytm, zastosowany na rezultat binaryzacji DSN, powoduje uzyskanie lepszych wyników w porównaniu do zastosowania samej binaryzacji (Rysunek 47 (b) i Rysunek 48 (b)). Pozwolił m.in. pozbyć się niewielkich nieciągłości. Jednak eksperymenty przeprowadzone na pełnym zbiorze obrazów testowych z wykorzystaniem miary dokładności nie pokazały znaczącej poprawy. Wynika to z faktu, że średni wynik uzyskany z DSN sam w sobie miał wysoką skuteczność. Ponadto algorytm postbinaryzacji mógł spowodować niewielkie powiększenie powierzchni obiektu.

6 Podsumowanie i wnioski

Celem niniejszej rozprawy doktorskiej było udowodnienie tezy, że istnieje uniwersalne podejście do preprocessnigu obrazów zawierających obiekty o liniowej strukturze, które poprawiałoby wyniki zwracane przez binaryzację. Został zaprezentowany nowy sposób na polepszanie jakości zbinaryzowanego obrazu. Zaproponowane podejście przedstawia kroki potrzebne, by ulepszyć obraz, nie narzucając przy tym konkretnych algorytmów, a jedynie oczekiwany rezultat wykonania operacji. Wyniki pokazują, że algorytm poprawnie współpracuje z różnymi metodami binaryzacji. Ponadto nie narzuca konkretnej metody wyznaczania pola kierunkowego, polepszając obraz wejściowy zarówno dla podejścia z gradientem, jak i z hesjanem. Zaproponowano także miarę dokładności, która udowadnia skuteczność działania przedstawionej metody.

Rezultaty testów wskazują, że zaproponowany algorytm wpływa pozytywnie na wyniki binaryzacji. Tworzy obraz, który jest bardziej podobny do oczekiwanego, idealnego rezultatu. W szczególności, część algorytmu polegająca na podwojeniu rozmiaru obrazu znacząco poprawia efekt końcowy. Zaskakującym może być fakt, że metoda oparta na obliczeniu Hesjanu, której autorami są Frangi et al. [31], nie osiąga lepszych wyników w stosunku do metody opartej na gradiencie.

Przedstawiony algorytm potrafi efektywnie zredukować liczbę artefaktów powstałych w procesie binaryzacji. Ponadto pozwala naprawić uszkodzone linie, jednocześnie nie pogrubiając tekstu, dzięki czemu nie usuwa niewielkich pętli. Warto zaznaczyć, że metoda może zostać zastosowana do dowolnego algorytmu binaryzacji, pod warunkiem, że ma ona dostęp także do obrazu w odcieniach szarości. Jest to wymagane do obliczenia pola kierunkowego. Istotną obserwacją jest fakt, że dla tych samych parametrów, w zależności od wybranej techniki binaryzacji, wyniki zostaną poprawione w stosunku do obrazu bazowego, jednak będą się różnić między sobą. Zatem przed zastosowaniem algorytmu postbinaryzacyjnego, warto przeanalizować, który algorytm binaryzacji pozwala osiągnąć najlepszy rezultat w danym problemie.

Warto zwrócić uwagę, że opisywany algorytm skutecznie udoskonala obraz wejściowy i nie powoduje pogorszenia jego czytelności. Może być stosowany w istniejących algorytmach wykorzystujących binaryzację, jako dodatkowy krok usprawniający działanie. W efekcie jego działania powinien zostać otrzymany obraz, zawierający obiekt, którego cechy są łatwiejsze do wyekstrahowania. To w konsekwencji przekłada się na skuteczniejszą klasyfikację. W przypadku prostych problemów, takich jak baza MNIST [52], algorytmy preprocessingu nie oferują znaczących udoskonaleń. Jednak w przypadku bardziej rzeczywistych problemów, takich jak

zagniecenia papieru, plamy, nierówne oświetlenie czy częściowo rozmazany obraz, preprocessing skutecznie pomaga poprawiać jakość obrazu. Takie przypadki zostały przedstawione w bazie DIBCO [62]. Co więcej, algorytmy preprocessingu poprawiają skuteczność całego procesu rozpoznawania obrazów także w przypadku zdjęć pozbawionych tak zróżnicowanych błędów, a zawierających tekst z dużą ilością szumów ([5]).

Przedstawiony algorytm jest w stanie poprawić efekt binaryzacji, jednak warto zwrócić uwagę na jego zachowanie w przypadku operowania na obrazach o niskiej liczbie niedoskonałości. Jak wykazano w rozdziale 5.4, zastosowanie filtrów, które skutecznie naprawiają zerwane połączenia między liniami może skutkować pogrubieniem całego obiektu. W niektórych przypadkach nie jest to problemem w kontekście przetwarzania takiego obrazu w dalszych etapach, w tym klasyfikacji. Jeśli pogrubienie poszczególnych glifów nie powoduje utraty informacji, taka operacja powinna być neutralna z punktu widzenia dalszej analizy. Jednak niniejsza praca prezentuje przypadki, w których zwiększenie objętości poszczególnych znaków wiąże się z negatywnymi konsekwencjami. Jest to szczególnie dotkliwe w przypadku obrazów o niskiej rozdzielczości. Przykładowo, usunięta w ten sposób pętla może być cechą charakterystyczną danego pisarza. Złączone w ten sposób dwie osobne linie mogą skutkować brakiem możliwości zidentyfikowania dwóch oddzielnych liter. Kolejnym problemem może być złączenie akcentu glifu z jego zasadniczą częścią. Może to mieć miejsce w przeanalizowanej bazie PBOK [5]. Taki błąd uniemożliwia identyfikację akcentu, co w konsekwencji może skutkować rozpoznaniem zupełnie innego słowa niż to, które zostało zapisane.

Prezentowany algorytm może być zastosowany w wielu praktycznych sytuacjach. Jednym z dużych obszarów jest digitalizacja historycznych dokumentów. Takie pisma mogą używać alfabetu różnego od łacińskiego. Jak zostało wykazane w niniejszej rozprawie, prezentowany algorytm jest w stanie efektywnie przetworzyć dokumenty historyczne, co nie jest regułą dla innych algorytmów. Warto zwrócić szczególną uwagę na starodawne teksty napisane ozdobną czcionką. Zawierają one wiele pętli i zdobień. Niepoprawny preprocessing takiego dokumentu może uniemożliwić identyfikację poszczególnych liter. W [43] zwrócono uwagę, że problem jest na tyle skomplikowany, że warto wykorzystywać do niego półautomatyczne systemy. Dwa najefektywniejsze systemy, GIDOC oraz MM-CATTI, bazują na wiedzy specjalistów, na podstawie których system uczy się poprawnej klasyfikacji. Innym problemem, do którego może zostać wykorzystany prezentowany algorytm, jest rozpoznawanie pisma w księgach małżeństw. Takie dokumenty są zwykle uzupełniane ręcznie. Problem rozpoznawania pisma jest o tyle bar-

dziej zaawansowany, że zawiera nazwiska - słowa, które nie muszą istnieć w słowniku. To sprawia, że rozpoznanie pełnego wyrazu może być problematyczne także dla człowieka. Jednym z rozwiązań tego problemu jest wykorzystanie wiedzy a priori [79]. Dzięki znajomości struktury takich ksiąg, można przewidzieć miejsca, w których dane nazwisko jest powtarzane. Przy wystarczająco skutecznym algorytmie preprocessingu, czytelnik ma możliwość identyfikacji nazwiska na podstawie wielu próbek, co powinno uczynić problem nieco mniej skomplikowanym. Kolejnym problemem, do którego może zostać wykorzystany prezentowany algorytm jest digitalizacja dokumentów bankowych. Systemy OCR są w tym celu bardzo szeroko wykorzystywane i dla alfabetu łacińskiego dają zadowalające rezultaty. Niestety, w przypadku innych alfabetów praktyczność takich systemów drastycznie spada. W [65] Autorzy poruszają problem dokumentów bankowych w języku batak. Zwracają uwagę, że istniejące systemy, działające na alfabecie łacińskim w tym przypadku nie dają satysfakcjonujących wyników. W publikacji zaproponowano metodę, której skuteczność różni się w zależności od koloru i faktury podłoża, na którym został napisany tekst. Algorytm prezentowany w niniejszej pracy skutecznie radzi sobie z problemem plam, zagięć czy prześwitującego tekstu z drugiej strony. Mógłby więc poprawić wyniki klasyfikatora.

Zastosowanie algorytmu nie musi ograniczać się jedynie do pisma ręcznego. Może zostać użyte dla każdego problemu, w którym występują obiekty ze strukturą liniową. Innymi problemami, w których metoda znajduje zastosowanie to obrazy medyczne, płytki PCB, rozpoznawanie tablic rejestracyjnych czy filtry wideo w serwisach społecznościowych. Zastosowanie w przedstawionych kategoriach zostało szerzej opisane poniżej.

Dokładność w odwzorowaniu kształtów linii sprawia, że prezentowany algorytm może być stosowany także do obrazów medycznych. Literatura pokazuje, że na etapie preprocessingu oba typy algorytmów mogą być stosowane wymiennie. Podobieństwo obu typów algorytmów zostało przedstawione w rozdziale 5.4. Przykładem problemu medycznego, do którego może mieć zastosowanie prezentowany algorytm, mogą być systemy biometryczne, takie jak zdjęcia żył na dłoni [77] czy zdjęcia żył na gałkach ocznych [96]. Charakteryzują się one strukturą połączonych ze sobą linii, znajdujących się na niejednorodnym tle. Są to warunki, z którymi radzi sobie prezentowany algorytm, co zostało wykazane w niniejszej rozprawie. Takie zdjęcia mogą zostać wykorzystane do identyfikacji człowieka. Badania pokazują, że takie metody mogą oferować niemal bezbłędne rozpoznanie danej osoby [99]. Innym zagadnieniem z dziedziny biometrii, w którym zastosowanie ma prezentowany algorytm, jest identyfikacja człowieka na podstawie

linii papilarnych. Jest to funkcjonalność szeroko wykorzystywana m.in. w telefonach komórkowych. Implementacja takiego systemu może opierać się na binaryzacji obrazu z odciskiem linii papilarnych oraz zastosowaniu algorytmu szkieletyzacji [94] czy sieci neuronowych [97]. Zachowanie ciągłości linii i odpowiednich odstępów między nimi jest kluczowe do poprawnego rozpoznania człowieka. Prezentowany algorytm zwraca szczególną uwagę na te elementy, co czyni jego wykorzystanie zasadnym dla tego typu systemów.

Innym problemem, do którego algorytm może zostać wykorzystany to identyfikowanie nieprawidłowości na płytkach PCB [53]. Zaprezentowane podejście może zostać wykorzystane także do niektórych algorytmów wektoryzacji, które, na etapie segmentacji, identyfikują kształty, w celu dalszej redukcji złożoności tych segmentów.

Kolejnym dużym obszarem, do którego można wykorzystać przedstawiony algorytm, jest rozpoznawanie tablic rejestracyjnych. Wiele galerii handlowych, w tym warszawska Galeria Młociny, oferuje automatyczne skanowanie tablicy rejestracyjnej podczas wjeżdżania na parking. To sprawia, że ruch podczas wjazdu jest płynniejszy. Kierowca nie musi pobierać specjalnego biletu, opłata, za jego pobyt na parkingu dokonuje się od momentu zakończenia czytania danych o jego samochodzie. Podobne rozwiązanie jest stosowane na autostradach. Coraz więcej bramek autostradowych wymieniana jest na automatyczne systemy naliczające opłatę za przejazd. W tym przypadku skutkuje to nie tylko zwiększeniem płynności ruchu, ale także redukcją zatrudnienia. Automatyczny system poboru opłat nie wymaga od pracowników stałej obecności w tych miejscach. Istnieją systemy ingerujące dużo bardziej w zachowania kierowcy. Warto zwrócić uwagę na coraz prężniej rozwijające się rozwiązania VIRS (Vehicle Information Retrieval System). Są to systemy nadzorujące pojazdy na drodze. Przykładowo w [101] zaproponowano algorytm monitorujący i analizujący samochody poruszające się po danej drodze. Poprzez czytanie numeru rejestracyjnego pojazdu, algorytm zdobywa informacje na temat wieku pojazdu, jego stanu technicznego, ubezpieczenia czy rodzaju silnika. Na tej podstawie stara się przewidzieć potencjalne wypadki, a informacje o swoich predykcjach są następnie przekazywane do odpowiednich służb. Innym podejściem może poszczycić się miasto Szczecin, dla którego został opracowany system monitoringu zwiększający bezpieczeństwo [93]. W tym przypadku czytanie tablic rejestracyjnych ma na celu rozszerzenie możliwości policji i straży miejskiej. Autorzy zwracają uwagę na to, że człowiek jest bardziej zawodny od algorytmu i nie podejmuje decyzji tak szybko jak algorytm. Wykrycie potencjalnych kradzieży samochodów czy kierowców rażąco łamiących prawo często wymaga natychmiastowej reakcji. W takich

sytuacjach przewagę nad człowiekiem ma system automatyczny.

Innym obszarem, aktualnie dynamicznie rozwijającym się, są filtry wideo. Modyfikują one wyświetlany obraz, zazwyczaj powodując upiększenie osób bądź identyfikację i modyfikację pewnych elementów obrazu. W czasach pandemii ich zastosowanie wzrosło w aplikacjach pozwalających zdalną komunikację, takich jak Zoom czy Microsoft Teams. Każda z takich aplikacji pozwala zmodyfikować tło, wprowadzając tzw. wirtualne tło. Wszystkie te aplikacje bazują na identyfikacji linii będących konturem postaci. Cienkie linie konturu mogą zostać wzmocnione poprzez zastosowanie algorytmu preprocessingu zaprezentowanego w niniejszej rozprawie doktorskiej. Jest to szczególnie istotne w przypadku wideokonferencji. W [14] udowodniono, że wpływa to nie tylko na bezpieczeństwo (rozmówca nie jest w stanie zobaczyć wyglądu pokoju), lecz także na poziom skupienia. Jest on znacznie wyższy dzięki eliminacji rozpraszaczy.

Prezentowany algorytm wykazuje duży potencjał co do różnorodności zagadnień, w jakich może zostać wykorzystany. Każdy problem wymaga jednak ręcznego dostosowania parametrów do typu przetwarzanych obrazów. Kolejne badania nad algorytmem powinny skupiać się na zwiększeniu automatyzacji działania algorytmu. Taka automatyzacja mogłaby zakładać wstępną analizę obrazów przetwarzanej bazy w celu dostosowania parametrów algorytmu do tych obrazów. Wiele z tych parametrów jest zależnych od wymiarów identyfikowanych obiektów czy rozdzielczości obrazów. Wstępna analiza mogłaby mierzyć te wartości i proponować zestaw parametrów, spośród których użytkownik mógłby wybrać efekt, który odpowiada mu najbardziej.

Inną propozycją rozwinięcia algorytmu jest skupienie się na jego wydajności. Priorytetem prezentowanej metody jest precyzja i dokładność zwracanych wyników. W niektórych sytuacjach (takich jak filtry stosowane podczas wideokonferencji) taka precyzja jest mniej istotna w porównaniu do szybkości przetwarzania poszczególnych klatek filmu. Jako że prezentowany algorytm jest elastyczny i składa się z wielu komponentów, warto zastanowić się, jakie komponenty należałoby wybrać, aby zoptymalizować go pod względem wydajności.

Bibliografia

[1] Baza DIBCO (2017). *https://vc.ee.duth.gr/dibco2017 (dostęp 18.08.2021)*.

[2] Baza ICFHR (2012). *http://www.iapr-tc11.org/mediawiki/index.php?title=ICFHR_2012_Signature_Verification_Competition_(4NSigComp2012) (dostęp 18.08.2021)*.

[3] M. Adamski, K. Sarnacki, and K. Saeed. Binary handwriting image enhancement by directional field-guided morphology. *Information Sciences*, 551:168–183, 2021.

[4] Y. Akbari, S. Al-Maadeed, and K. Adam. Binarization of Degraded Document Images Using Convolutional Neural Networks and Wavelet-Based Multichannel Images. *IEEE Access*, 8, 2020.

[5] A. Alaei, U. Pal, and P. Nagabhushan. Dataset and ground truth for handwritten text in four different scripts. *International Journal of Pattern Recognition and Artificial Intelligence*, 2012.

[6] N. Altwaijry and I. Al-Turaiki. Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 33, 2021.

[7] B. Amirgaliyev, M. Kairanbay, and K. Chingiz. Effective algorithms and methods for automatic number plate recognition. pages 1–4, 2014.

[8] S. Anitha and V. Radha. Comparison of Image Preprocessing Techniques for Textile Texture Images. *International Journal of Engineering Science and Technology*, 2, 2010.

[9] M. A. Ansari, S. Zai, and Y. S. Moon. Performance Comparison of Vesselness Measures for Segmentation of Coronary Arteries in 2D Angiograms. *Indian Journal of Science and Technology*, 9(48), 2016.

[10] R. Bajaj and S. Chaudhury. Signature verification using multiple neural classifiers. *Pattern Recognition*, 30(1):1–7, 1997.

[11] A. M. Bazen and S. H. Gerez. Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):905–919, 2002.

[12] M. Bessmeltsev and J. Solomon. Vectorization of Line Drawings via Polyvector Fields. *ACM Trans. Graph.*, 38(1):9:1–9:12, 2019.

[13] C. A. Boiangiu, I. Bucur, and A. Tigora. The image binarization problem revisited: perspectives and approaches. *Journal of Information Systems Operations Management*, 6:419–427, 2012.

[14] M. Boyle, C. Edwards, and S. Greenberg. The Effects of Filtered Video on Awareness and Privacy. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 1–10, 2000.

[15] V. Braverman. Sliding Window Algorithms. *Encyclopedia of Algorithms*, pages 2006–2011, 2016.

[16] W. Burger and M. Burge. Color Quantization. pages 329–339, 2016.

[17] N. Chaki, S. H. Shaikh, and K. Saeed. Exploring Image Binarization Techniques. *Springer Publishing Company, Incorporated*, 2014.

[18] Y. Chen and L. Wang. Broken and degraded document images binarization. *Neurocomputing*, 237:272–280, 2017.

[19] M. Coşkun, Ö. Yıldırım, A. Uçar, and Y. Demir. An overview of popular deep learning methods. *European Journal of Technic*, 7:165–176, 2017.

[20] C. Cuesta, M. Ruesta, D. Tuesta, and P. Urbiola. The digital transformation of the banking industry. *BBVA Research Digital Watch*, 2015.

[21] J. Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[22] P. A. Devijver and J. Kittler. Pattern Recognition: A Statistical Approach. *Prentice/Hall International*, 1982.

[23] B. Dhanddra, V. Malemath, M. Hangarge, and R. Hegadi. Multi-font English Character Recognition based on Modified Invariant Moments. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 67:153–162, 2008.

[24] M. Diaz-Cabrera, A. Morales, and M. A. Ferrer. Emerging issues for static handwritten signature biometrics. *Advances in Digital Handwritten Signature Processing*, pages 109–120, 2014.

[25] M. Diem and R. Sablatnig. Recognizing Degraded Handwritten Characters. *CVL Technical Report*, 2010.

[26] L. Donati, S. Cesano, and A. Prati. A complete hand-drawn sketch vectorization framework. *Multimedia Tools and Applications*, 78(14):19083–19113, 2019.

[27] V. Egiazarian, O. Voynov, A. Artemov, D. Volkhonskiy, A. Safin, M. Taktasheva, D. Zorin, and E. Burnaev. Deep Vectorization of Technical Drawings. *Computer Vision – ECCV 2020*, pages 582–598, 2020.

[28] F. Fardian, F. Arnia, S. Muchallil, and K. Munadi. Identification of most suitable binarisation methods for acehnese ancient manuscripts restoration software user guide. *Jurnal Teknologi*, 77, 2015.

[29] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordez. Robustness of Offline Signature Verification Based on Gray Level Features. *IEEE Transactions on Information Forensics and Security*, 7(3):966–977, 2012.

[30] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez, and J. Ortega-Garcia. An Off-line Signature Verification System Based on Fusion of Local and Global Information. *Biometric Authentication*, 3087:295–306, 2004.

[31] A. Frangi, W. Niessen, K. Vinc, and M. Viergever. Multiscale Vessel Enhancement Filtering. *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98*, 1496:130, 1998.

[32] E. Gedraite and M. Hadad. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. *International Symposium on Electronics in Marine, 2011 Proceedings*, pages 393–396, 2011.

[33] R. C. Gonzalez and R. E. Woods. Digital Image Processing (3rd Edition). 2007.

[34] K. V. Greeshma and K. Sreekumar. Fashion-MNIST classification based on HOG feature descriptor using SVM. *International Journal of Innovative Technology and Exploring Engineering*, 8:960–962, 2019.

[35] Y. Guo, C. Ji, X. Zheng, Q. Wang, and X. Luo. Multi-scale Multi-attention Network for Moiré Document Image Binarization. *Signal Processing: Image Communication*, 90, 2021.

[36] M. R. Gupta, N. P. Jacobson, and E. K. Garcia. OCR binarization and image pre-processing for searching historical documents. *Pattern Recognition*, 40(2):389–397, 2007.

[37] S. He and L. Schomaker. DeepOtsu: Document enhancement and binarization using iterative deep learning. *Pattern Recognition*, 2019.

[38] L. Hong, Y. Wan, and A. Jain. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.

[39] K. Huang and H. Yan. Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognition*, 30(1):9–17, 1997.

[40] D. Impedovo and G. Pirlo. Automatic Signature Verification: The State of the Art. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 38(5):609–635, 2008.

[41] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal. Offline Recognition of Devanagari Script: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews*, 41(6):782–796, 2011.

[42] F. Jia, C. Shi, K. He, C. Wang, and B. Xiao. Degraded document image binarization using structural symmetry of strokes. *Pattern Recognition*, 74:225–240, 2018.

[43] A. Juan, V. Romero, J.A. Sánchez, N. Serrano, A.H. Toselli, and E. Vidal. Handwritten Text Recognition for Ancient Documents. *Journal of Machine Learning Research - Proceedings Track*, 11:58–65, 2010.

[44] H. Karwal and A. Girdhar. Vehicle Number Plate Detection System for Indian Vehicles. *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, pages 8–12, 2015.

[45] S. Kaur and S. Kaur. An Efficient Approach for Number Plate Extraction from Vehicles Image under Image Processing. *International Journal of Computer Science and Information Technologies*, 5:2–11, 2014.

[46] D. J. Kennard, W. A. Barrett, and T. W. Sederberg. Offline Signature Verification and Forgery Detection Using a 2-D Geometric Warping Approach. *21st International Conference on Pattern Recognition (ICPR), Tsukuba, Japan*, pages 3733–3736, 2012.

[47] R. Keys. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153, 1981.

[48] P. Khaire and N. Thakur. An Overview Of Image Segmentation Algorithms. *International Journal of Image Processing and Vision Sciences*, 1:62–68, 2011.

[49] C. Kirbas and F. Quek. A Review of Vessel Extraction Techniques and Algorithms. *ACM Computing Surveys*, 36, 2002.

[50] H. Krottmaier. The Future of Digital Libraries. 2004.

[51] C. Kusuma, A. Fadhilah, and A. Afiahayati. Convolutional Neural Networks for Handwritten Javanese Character Recognition. *Indonesian Journal of Computing and Cybernetics Systems*, 12, 2018.

[52] Y. LeCun and C. Institute. THE MNIST DATABASE of handwritten digits. 1998.

[53] D. Li, S. Li, and W. Yuan. Flexible Printed Circuit Fracture Detection Based on Hypothesis Testing Strategy. *IEEE Access*, 8:24457–24470, 2020.

[54] J. Liu. The Influence of the Body Image Presented Through TikTok Trend-Videos and Its Possible Reasons. 2021.

[55] L.M. Lorigo and V. Govindaraju. Offline Arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):712–724, 2006.

[56] A. P. Mahmoudzadeh and N. Kashou. Evaluation of Interpolation Effects on Upsampling and Accuracy of Cost Functions-Based Optimized Automatic Image Registration. *International journal of biomedical imaging*, 2013.

[57] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. Handbook of Fingerprint Recognition. 2009.

[58] P. Manwatkar and K. Singh. A Technical Review on Text Recognition from Images. 2015.

[59] F . D. Martino and S. Sessa. PSO image thresholding on images compressed via fuzzy transforms. *Information Sciences*, 506:308–324, 2020.

[60] S. Mathur, V. Aggarwal, H. Joshi, and A. Ahlawat. Offline Handwriting Recognition Using Genetic Algorithm. *Sixth International Conference on Information Research and Applications – i.Tech 2008*, 2008.

[61] M. Mcauliffe, F. Lalonde, D. P. McGarry, W. Gandler, K. Csaky, and B. Trus. Medical Image Processing, Analysis Visualization in Clinical Research. *Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems*, 14:381 – 386, 2001.

[62] H. Michalak and K. Okarma. Improvement of Image Binarization Methods Using Image Preprocessing with Local Entropy Filtering for Alphanumerical Character Recognition Purposes. *Entropy*, 21:562, 2019.

[63] N. Mitianoudis and N. Papamarkos. Multi-spectral document image binarization using image fusion and background subtraction techniques. *IEEE International Conference on Image Processing (ICIP)*, pages 5172–5176, 2014.

[64] A. Molder, O. Martens, and T. Saar. Adaptively undersampled, circular histogram based image processing for rotation invariant coin detection. *13th Biennial Baltic Electronics Conference*, 2012.

[65] M. Muchtar, I. Jaya, N. Siregar, E. Nababan, S. Effendi, O. Sitompul, M. Zarlis, U. Andayani, T. Nasution, I. Siregar, and M. Syahputra. Digitization of Batak Manuscripts Using Methods Learning Vector Quantization (LVQ). *IOP Conference Series: Materials Science and Engineering*, 851, 2020.

[66] R. P. Mundani, J. Frisch, V. Varduhn, and E. Rank. A sliding window technique for interactive high-performance computing scenarios. *Advances in Engineering Software*, 84:21–30, 2015.

[67] D. Muramatsu and T. Matsumoto. Effectiveness of Pen Pressure, Azimuth, and Altitude Features for Online Signature Verification. *Advances in Biometrics*, pages 503–512, 2007.

[68] W. Niblack. An Introduction to Digital Image Processing. *Prentice Hall, Englewood Cliffs*, 1986.

[69] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[70] J. Pastor-Pellicer, S. España-Boquera, F. Zamora-Martínez, M. Z. Afzal, and M. J. Castro-Bleda. Insights on the Use of Convolutional Neural Networks for Document Image Binarization. *Advances in Computational Intelligence*, pages 115–126, 2015.

[71] D. Pedamonti. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. 2018.

[72] C. Pellot, A. Herment, M. Sigelle, P. Horain, H. Maitre, and P. Peronneau. A 3D reconstruction of vascular structures from two X-ray angiograms using an adapted simulated annealing algorithm. *IEEE Trans Med Imaging*, 1994.

[73] P. Prabhakar, P. Anupama, and S. Resmi. Automatic vehicle number plate detection and recognition. *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, pages 185–190, 2014.

[74] S. B. Pratama. SVD Implementation on MNIST Image Classification Based on CNN. 2021.

[75] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1395–1403, 2017.

[76] K. Puloria and S. Mahajan. A Review on Automatic Number Plate Recognition System. *International Journal of Software and Hardware Research in Engineering*, 3(1), 2015.

[77] H. Qin, X. He, X. Yao, and H. Li. Finger-vein verification based on the curvature in Radon space. *Expert Systems with Applications*, 82:151–161, 2017.

[78] J. Richiardi, H. Ketabdar, and A. Drygajlo. Local and Global Feature Selection for Online Signature Verification. *Eighth International Conference on Document Analysis and Recognition*, (625-629), 2005.

[79] V. Romero, J.A. Sánchez, N. Serrano, and E. Vidal. Handwritten Text Recognition for Marriage Register Books. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 533–537, 2011.

[80] K. Saeed, M. Tabedzki, M. Rybnik, and M. Adamski. K3M: A universal algorithm for image skeletonization and a review of thinning techniques. *Applied Mathematics and Computer Science*, 20:317–335, 2010.

[81] U. Salimah, V. Maharani, and R. Nursyanti. Automatic License Plate Recognition Using Optical Character Recognition. *IOP Conference Series: Materials Science and Engineering*, 1115, 2021.

[82] K. Sarnacki, M. Adamski, and K. Saeed. Signature Image Improvement with Gradient Adaptive Morphology. *International Conference on Computer Information Systems and Industrial Management Applications*, pages 50–58, 2019.

[83] K. Sarnacki and K. Saeed. Character Recognition Based on Skeleton Analysis. *International Conference, ICCVG 2018*, pages 148–159, 2018.

[84] J. Sauvola and M. Pietikinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[85] G. N. Sazaklis. Geometric Methods for Optical Character Recognition. *Rozprawa doktorska, State University of New York at Stony Brook*, 1997.

[86] A. Sehad, Y. Chibani, R Hedjam, and M. Cheriet. Gabor filter based texture for ancient degraded document image binarization. *Pattern Analysis and Applications*, 22(1):1–22, 2019.

[87] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004.

[88] A. Sharko, E. Meka, G. Sharko, and I. Baholli. Digital Banking the Wave of the Future. *ISTI 2016: the New Paradigm for a Smarter Economy*, 2017.

[89] V. Shrivastava and N. Sharma. Artificial Neural Network Based Optical Character Recognition. *Signal Image Processing : An International Journal*, 3, 2012.

[90] K. Sonavane, B. Soni, and U. Majhi. Survey on Automatic Number Plate Recognition (ANR). *International Journal of Computer Applications*, 125:1–4, 2015.

[91] R. Srisha and A. Khan. Morphological Operations for Image Processing : Understanding and its Applications. *National Conference on VLSI, Signal processing Communications*, 2013.

[92] J. Stastny and M. Minařík. A Brief Introduction to Image Pre- Processing for Object Recognition. *ICSC - International Conference on Soft Computing Applied in Computer and Economic Enviroment*, 2007.

[93] S. Stempiński. Image monitoring as a factor shaping the security of the municipality of the city of Szczecin. *Modern Management Review*, XXIV, 2019.

[94] K. Surmacz and K. Saeed. Robust Algorithm for Fingerprint Identification with a Simple Image Descriptor. *Computer Information Systems - Analysis and Technologies*, pages 137–144, 2011.

[95] G. T. Sutar and A. V. Shah. Number Plate Recognition Using an Improved Segmentation. *International Journal of Innovative Research in Science*, 3(5), 2014.

[96] M. Szymkowski, D. Najda, and K. Saeed. An Algorithm for Exact Retinal Vein Extraction. *Computer Information Systems and Industrial Management*, pages 72–83, 2019.

[97] M. Szymkowski and K. Saeed. Fingerprint Feature Extraction with Artificial Neural Network and Image Processing Methods. *Advances in Soft and Hard Computing*, pages 86–97, 2019.

[98] C. Tensmeyer and T. Martinez. Document Image Binarization with Fully Convolutional Neural Networks. *14th IAPR International Conference on Document Analysis and Recognition*, pages 99–104, 2017.

[99] S. Tuama and L. George. Automatic Human Recognition Based on the Geometry of Retinal Blood Vessels Network. *Current Research in Science and Technology*, pages 1–17, 2019.

[100] A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hildebrandt, and M. Ben-Chen. Directional Field Synthesis, Design, and Processing. *Computer Graphics Forum*, 35:545–572, 2016.

[101] R. Venkatesh. Vehicle Information Retrieval by Number Plate Detection. 2020.

[102] R. Verd-Monedero, J. Angulo, and J. Serra. Anisotropic Morphological Filters With Spatially-Variant Structuring Elements Based on Image-Dependent Gradient Fields. *IEEE Transactions on Image Processing*, 20(1):200–212, 2011.

[103] G. D. Vo and C. Park. Robust regression for image binarization under heavy noise and nonuniform background. *Pattern Recognition*, 81:224–239, 2018.

[104] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee. Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, 74:568–586, 2018.

[105] A. Wronska, K. Sarnacki, and K. Saeed. Athlete number detection on the basis of their face images. *2017 International Conference on Biometrics and Kansei Engineering*, pages 84–89, 2017.

[106] L. Wu, P. Jiang, Z. C. Chen, X. Lin, Y. Lai, P. Lin, and S. Cheng. BitFlow-Net: Toward Fully Binarized Convolutional Neural Networks. *IEEE Access*, 2019.

[107] L. Xu and S. Luo. A novel method for blood vessel detection from retinal images. *Biomedical engineering online*, 9:14, 2010.

[108] Q. Ye, S. Jiang, Y. Liu, and W. Gao. Jersey number detection in sports video for athlete identification. *Proceedings of SPIE - The International Society for Optical Engineering*, 5960:1599–1606, 2005.

[109] A. Youn. What Is the Ideal Instagram Filter? *Aesthetic Surgery Journal Open Forum*, 1, 2019.