**Oprogramowanie OCR w zastosowaniu praktycznym**

Oleksandr Mertsalov

**Spis treści**

[Wstęp………………………………………………………………………………………….3](#_Toc72871887)

[Rozdział I](#_Toc72871888)

[**Optyczne rozpoznawanie znaków 4**](#_Toc72871889)

[1.1. Sieci neuronowe 7](#_Toc72871890)

[1.2. Long short-term memory (LSTM) 7](#_Toc72871891)

[1.3. Tesseract 7](#_Toc72871892)

[Rozdział II](#_Toc72871893)

[Rozdział III](#_Toc72871894)

[Podsumowanie 7](#_Toc72871895)

[Bibliografia 7](#_Toc72871896)

## Wstęp

Obecnie coraz więcej dokumentów przechowujemy w sposób elektroniczny. Od 2020 roku pojawiła się nawet możliwość załatwienia spraw urzędowych przez Internet i składanie wniosków bez wychodzenia z domu. Trzymanie dokumentów na komputerze ubezpiecza nas od możliwej straty informacji z powodu fizycznych szkód, a w przypadku ich dużej ilości wielką zaletą jest szybkość, podczas sortowania, filtracji i wyszukiwania danych. Danymi, które te dokumenty mogą zawierać, są tekst i obrazy. Tekst - jest zbiorem zdań, zdania jest zbiorem słów, a słowo składa się z liter.

Wprowadzanie tekstu, zawartego w dokumencie do komputera, odbywa się za pomocą klawisz klawiatury, każda z których jest oznakowana pojedynczymi literami, cyframi lub symbolami interpunkcyjnymi. Komputer traktuje wciśnięta klawisze jako unikalny kod binarny lub liczbowy, a następnie używa wybranego przez system pliku czcionki, w celu wyświetlenia powiązanego z nim znaku na ekranie monitora. Stworzenie elektronicznej kopii dokumentu poprzez ręczne wpisywania każdego znaku jest dość czasochłonne w przypadku, gdy składa się on z więcej niż jednej strony, ale takie rozwiązanie powoduje, że w wyniku dostajemy stworzony edytowalny tekst.

Szybszym sposobem na przekazanie dokumentu komputerowi jest stworzenie jego wersji cyfrowej za pomocą skanera lub kamery. Taki dokument zostanie umieszczony w pamięci komputera jako plik graficzny, zapisany w odpowiednim formacie, najczęściej którymi są: JPEG, PDF, PNG albo TIFF. Tego typu plik przedstawia obraz na ekranie za pomocą zawartej informacji o ilości pikseli oraz ich położenia i koloru. Z tego wynika, że taka wersja dokumentu elektronicznego, służy tylko do podglądu jego treści.

Od lat 1950 ciągle się rozwija rozwiązanie, za pomocą którego w dzisiejszych czasach można uzyskać tekst z każdego pliku graficznego. Jest to oprogramowanie komputerowe, które się nazywa „OCR”. W swojej pracy pokażę jego zastosowanie we własnej aplikacji, cel której, jest odczytywanie danych z najczęściej spotykanych wśród ludzi dokumentów. Jest to dokument potwierdzający sprzedaż – „Paragon Fiskalny”.

# Rozdział I

## Optyczne rozpoznawanie znaków

Optyczne rozpoznawanie znaków (Optical Character Recognition, OCR), jest to technologia, która automatycznie identyfikuję liczby, litery oraz znaki interpunkcyjne w plikach graficznych i konwertuje ich do postaci danych, zrozumiałych komputerowi. Na wejściu podawany jest wcześniej przygotowany plik rastrowy, który poddaje się analizie, w celu odnajdywania w nim zbioru pikseli, wizualizujących symbol, a następnie do każdego znalezionego, przypisuje się kod binarny lub liczbowy z tablicy znaków. W wyniku, zwracany jest stworzony, edytowalny tekst. Przy użyciu skanera lub urządzenia wyposażanego w kamerę, rozpoznawanie tekstu może odbyć się w trakcie generowania pliku graficznego. W innym przypadku, możliwe jest przekazanie przygotowanej wcześniej grafiki, zawierającej tekst, do systemu OCR.

Obecnie nie istnieję idealnego oprogramowania, które zawsze potrafi rozpoznać tekst z dokładnością do 100%. Prawdopodobieństwo tego, że OCR odniesie sukces podczas pracy, zależy w pierwszej kolejności od jakości materiału wejściowego. Im większa jest rozdzielczość obrazu i jego kontrastowość, tym większa szansa, że tekst zostanie poprawnie rozpoznany. Drugim ważnym czynnikiem jest samo oprogramowanie, składające się z po kolei wywoływanych algorytmów służących do poprawy obrazu, segmentacji tekstu, rozpoznania znaków i analizie wyniku. Dziś można zaobserwować dużą ilość gotowych rozwiązań, które używają własne zaimplementowane sieci neuronowe, za pomocą których dokładność rozpoznawania znaków zrasta. Taki program jest uniwersalny, ponieważ pojawia się możliwość dostosować go do odpowiedniego problemu, i nauczyć rozumieć pismo ręczne lub tekst o dowolnej czcionce.

Diagram

Description automatically generatedW celu wyodrębnienia ciągu znakowego z obrazu, używane są różne techniki. Każda technika składa się z procesów, na wejście do których jest podawany wynik poprzedniego kroku, zaczynając od pliku graficznego i kończąc sformatowanym, edytowalnym tekstem. Przebieg pracy programu OCR jest zamieszczony na poniższym schemacie:

Rysunek 1.https://www.geeksforgeeks.org/tesseract-ocr-with-java-with-examples/

Praca systemów OCR w większości wypadków jest podzielona na cztery etapy:

1. **Przetwarzanie wstępne (ang. Pre-processing)**

Przekazany do OCR obraz, po procesie skanowania może zawierać pewną ilość szumów. Plamy, brud, zagięty lub pomarszczony papier, uszkodzona matryca urządzenia – jest to czynniki powodujące pojawienie w pliku rastrowym nie potrzebnych pikseli, utrudniających rozpoznawanie. Za duży poziom jasności, słaba jakość lub kontrast spowodują zniszczenie kształtu znaków, lub ich zamazanie. Dlatego, przed procesem identyfikacji znaków, na etapie wstępnym, implementuje się do obrazu metody programowe, które mają na celu polepszyć jakość pliku rastrowego oraz usunąć defekty, co spowoduję zwiększenie współczynnika rozpoznawania tekstu.

Obraz – jest zbiorem pikseli o różnych kolorach, aby rozgraniczyć, gdzie znajdują się w nim znaki, oprogramowanie OCR przyjmuję założenie, że są oni zawsze białego koloru, a wszystko pozostałe jest czarne. W fazie przetwarzania wstępnego, do zamiany kolorów na skalę szarości używany jest algorytm nazywany binaryzacja (z ang. binarization). Jego działanie wygląda następująco:

- przyjmuje się, że piksel białego koloru jest równy wartości 255, a czarny = 0. Następnie, odbywa się zaokrąglenie każdego pikselu według ustawionego progu (domyślne jest to połowa wartości 255, czyli 127). Jeśli wartość pikselu jest mniejsza od wartości progu, to taki piksel jest uznawany za czarny, w przeciwnym przypadku jego kolor zmieni się na biały.

1. **Segmentacja tekstu (ang. Text segmentation)**

Zanim program zacznie rozpoznawać znaki, istotnie jest najpierw rozbić zlokalizowany na obrazie tekst na pojedyncze linie, a te na słowa oraz litery. Dla tego celu, można użyć techniki histogramowej projekcji, która polega na wyliczaniu rzutów obrazu.

Najpierw, wykonuje się segmentacja linii tekstu, aby podzielić obszar tekstowy na pojedyncze linie. Odbywa się to za pomocą metody do wyliczania rzutu poziomowego (ang. Horizontal Histogram Projection). Ta metoda przyjmuje na wejście obraz binarny otrzymany z poprzedniego kroku, w którym białe pikseli - są uznawane za ważną informację, a czarne - za tło. Następnie, oblicza się suma białych pikseli w każdym wierszu obrazu. Wynikiem działania metody jest lista o długości równej wysokości obrazu. Jeśli wartość elementu listy, jest duża pod względem pozostałych elementów, to znaczy, że jest to część linii tekstu. W przeciwnym wypadku, elementy o malej wartości mogą być uznane za obszary znajdujące się pomiędzy liniami tekstu i służyć jako wskaźniki na miejsca, gdzie należy podzielić obraz.

Kolejnym zadaniem jest segmentacja słów. Używa się do tego metoda wyliczania rzutu pionowego (ang. Vertical Histogram Projection), do której przekazywany jest wycięty w poprzednim kroku obraz, zawierający pojedynczą linię tekstu. Jej działanie jest podobne do metody wykonywanej podczas segmentacji linii tekstu, jedną różnicą jest to, że tutaj oblicza się suma białych pikseli w każdej kolumnie obrazu a nie wierszu, jak to było wcześniej. Wynik – przedstawia listę o długości równej szerokości obrazu, a jej elementy w zależności od posiadanej wartości, mogą być częścią słów – jeśli są o dużej wartości lub polem pomiędzy nimi – w przypadku wartości niskich. Odstęp pomiędzy słowami jest większy niż odstęp pomiędzy znakami, dla tego ważne jest ustawić próg jego szerokości, na podstawie którego będą wybierane kolumny, rozdzielające słowa.

Proces segmentacji znaków zależy od typu tekstu na obrazie. Jeśli znaki tworzące słowo są niezależne i nie łączą się pomiędzy sobą, to w takim wypadku używa się metoda z poprzedniego kroku segmentacji. W tym wypadku, wyliczanie pionowego rzutu odbywa się na podstawie obrazu, który zawiera pojedyncze słowo, składające się z ciągu znaków. Na wyjściu zwracana jest lista, z której należy wybrać kolumny o niskich wartościach, będące separatorami. W innym przypadku, jeżeli litery w tekście, są połączone przez ligaturę, tak jak to często jest w piśmie ręcznym, to wtedy proces segmentacji słów, jest wykonywany przy pomocy sztucznej inteligencji.

1. **Rozpoznawanie znaków (ang. Character recognition)**

Jak tylko wszystkie potencjalne znaki znajdujące się na obrazu zostaną wyodrębnione, oprogramowanie OCR, może użyć jednej z dwóch technik do ich identyfikacji:

1. **rozpoznawanie wzorów (ang pattern recognition)** – polega na porównywaniu kształtów znajdujących się na obrazie z zestawem glifów w pamięci programu. Stosowanie tej techniki zaczęło się w latach 1960, do rozpoznawania dokumentów typu bankowych rachunków. Wtedy została stworzona czcionka o nazwie „OCR-A”, którą była używana podczas drukowania takich dokumentów. Zawierające w niej znaki, można było łatwo rozróżnić, a ich szerokość była jednakowa. To spowodowało, że współczynnik rozpoznawania tekstu dla tych czasów oprogramowania wzrósł niemal do 100%, a jego praca nie zajmowała dużo czasu. Następnym krokiem, było dodanie do systemów OCR, kolejnych wzorów do rozpoznawania innych popularnych czcionek.

Ta technika jest również znana jako „Dopasowywanie Macierzy (ang. Matrix Matching)”. Ta nazwa lepiej opisuje jej działanie, ponieważ obraz przetwarzany w tym kroku jest macierzą, gdzie czarne pikseli reprezentują wartość 1, a białe oznakowane jako 0. Przykład takiej macierzy jest zamieszczony poniżej:

Shape

Description automatically generated with low confidence

reference 1: <https://moov.ai/en/blog/optical-character-recognition-ocr/>

Rozpoznawanie zaczyna się od znalezienia centra macierzy. Następnie wylicza się promień, od jej centra do najbardziej oddalonej od niego jedynki za pomocą poniższej formuły:

*Dist = √ ((y2-y1) ² + (x2-x1) ²)*

Na podstawie wyliczonej długości promienia, macierz dzieli się na 5 równych obszarów, a te z kolei są dzielone na 8 segmentów. Końcowym wynikiem tego przetwarzania, jest nowa macierz o wymiarach [8x5]. Jej wartości są wyliczane na podstawie ilości jedynek w każdym odcinku. Taki sam format, mają szablony znajdujące się w pamięci programu.

Poniżej została umieszczona, ilustracja konwertowania binarnej wersji obrazu do postaci w jakiej są przechowywane szablony znaków:

Diagram

Description automatically generated

reference 2: TODO: PRZEROBIĆ na własne

Końcowym etapem, jest porównywanie wyniku z obecnymi w programie szablonami. Znak oznacza się zidentyfikowanym wtedy, gdy parametry porównywalnych macierze są do siebie podobne.

Ta technika dobrze radzi ze znanymi czcionkami, natomiast nie jest ona uniwersalna. W przypadku gdy znajdujący się na obrazie tekst jest pismem ręcznym, to jego rozpoznanie staję się niemożliwym.

1. **rozpoznawanie cech (ang. feature detection) –d.**

Минусом тут есть то, что этот способ не уверсальный, и не удастся распознать текст написаный шрифтом, который не знает програма.

1. **Post Processing**

## Sieci neuronowe

## Long short-term memory (LSTM)

## Tesseract

# Rozdział II

# Rozdział III

## Podsumowanie

## Bibliografia

* <https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7>
* <https://towardsdatascience.com/segmentation-in-ocr-10de176cf373>
* <https://www.explainthatstuff.com/how-ocr-works.html>
* <https://ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502254.pdf>