Metody Obliczeniowe w Nauce i Technice

Laboratorium 9

Wojciech Łącki

Spis treści

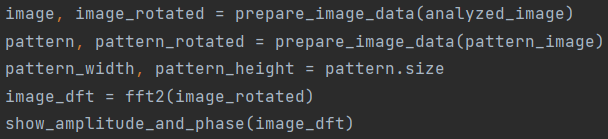
[Zadanie 1 – Analiza obrazów 2](#_Toc104328524)

[Zadanie 2 – OCR 5](#_Toc104328525)

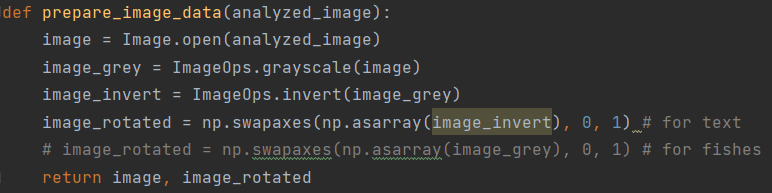
# Zadanie 1 – Analiza obrazów

1. Wczytaj obraz przedstawiający tekst, dokonaj odwrócenia kolorów (czarne tło), a następnie jego transformacji do domeny częstotliwościowej za pomocą DFT. Przedstaw wartości fazy i modułu współczynników Fouriera (osobno).

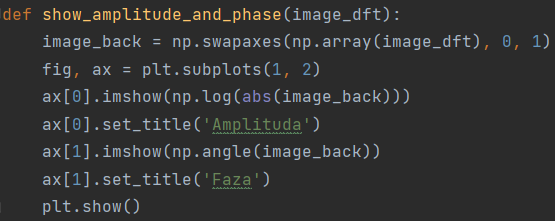
Kod, który wywołuje odpowiednie funkcje, aby otrzymać zamierzony efekt z zadania został przedstawiony poniżej.



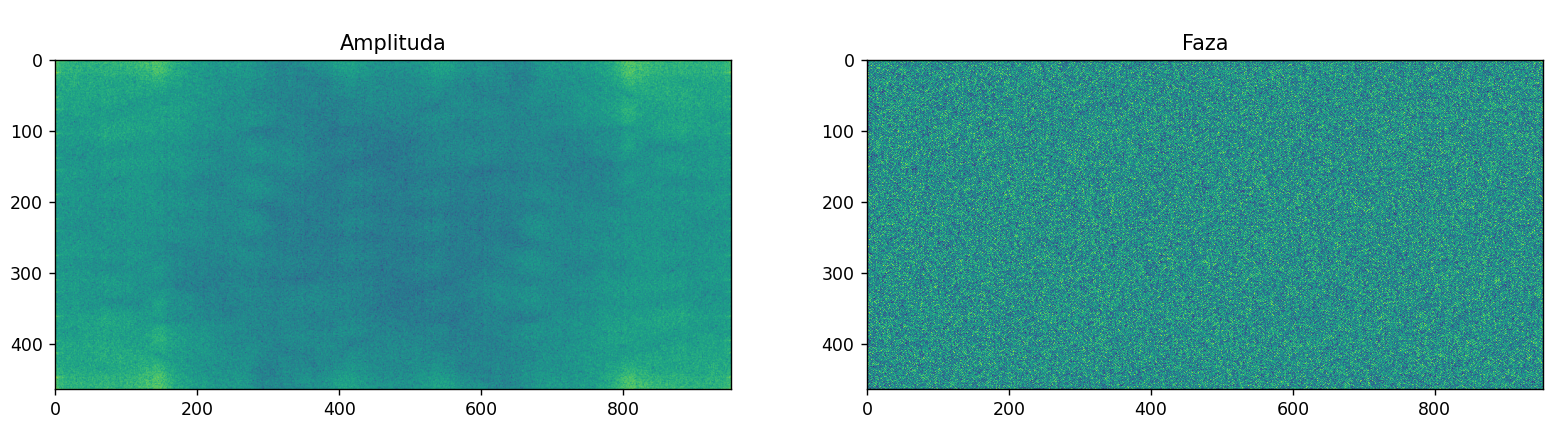
Funkcja przekształcająca otrzymany obraz dokonująca odpowiednich przekształceń obrazu, w tym zamiana kolorów i obrót. Dla tekstu zamieniamy kolory, natomiast dla ławicy nie (wtedy lepsze dpasowania).



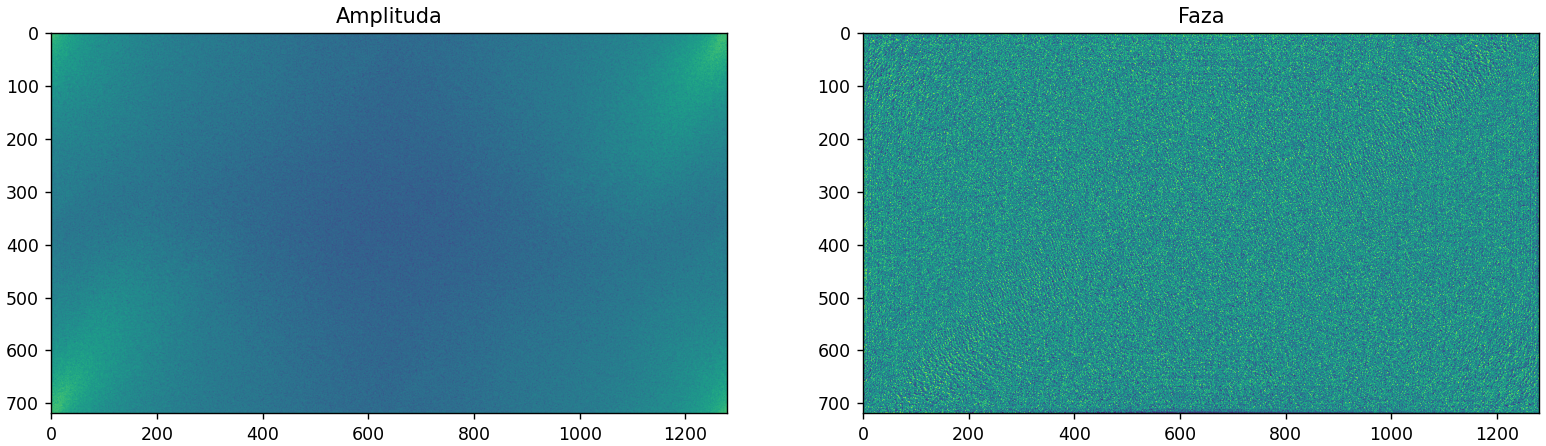
Natomiast faza i współczynniki można wyświetlić dzięki poniższej funkcji.



Wyniki uzyskane dla teksty.



Wyniki uzyskane dla ryb.



1. Wybierz przykładowego reprezentanta grupy - wytnij z dużego zdjęcia wzorzec obiektu   
   np. literę *e* lub charakterystyczny fragment ryby z ławicy (imcrop).

Dla tekstu została wycięta litera „e”.



Natomiast dla ławicy ryb fragment ryby.



1. Wykorzystując transformacje Fouriera oblicz korelacje między wybranym wzorcem, a całym obrazem.



1. Przedstaw otrzymany obraz wyjściowy oraz punkty, w których wykryto wystąpienie wzorca (nałóż je na obraz oryginalny).

Aby nałożyć wzorce na obraz, to najpierw trzeba wyfiltrować te współrzędne, w których należy je umieścić. Żeby to zrobić jednym z parametrów programu jest współczynnik, który mówi nam od jakiego pułapu zaczyna dana współrzędna być akceptowana.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Nałożenie wzorca wykonuje poniższy kod.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Pokazanie efektu nakładania wzorca.

Obraz zawierający tekst

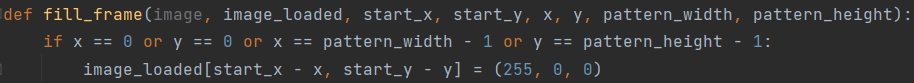
Opis wygenerowany automatycznie

Zostały napisane różne strategie zaznaczania obszarów, w których wykryto wzorzec.

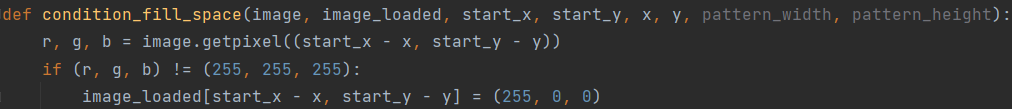
Pierwsza wypełnia całe miejsce.



Druga tylko ramkę.



Trzecia została stworzona tylko dla tekstu, żeby litery zmieniły kolor.



1. Znajdź liczbę wystąpień wzorca. Testy przeprowadź na dwóch zbiorach danych.

Dla tekstu otrzymaliśmy następującą liczbę wystąpień (współczynnik równy 0.9).



Dla ławicy było ich trochę więcej (współczynnik równy 0.75). Należy jednak zauważyć, że wyniki się pokrywają i nie jest to wykrywane, przynajmniej w tym zadaniu (przy OCR jest to już uwzględniane). Różnych wyników można zaobserwować 16, tak wynika patrząc na zbiory wyników.

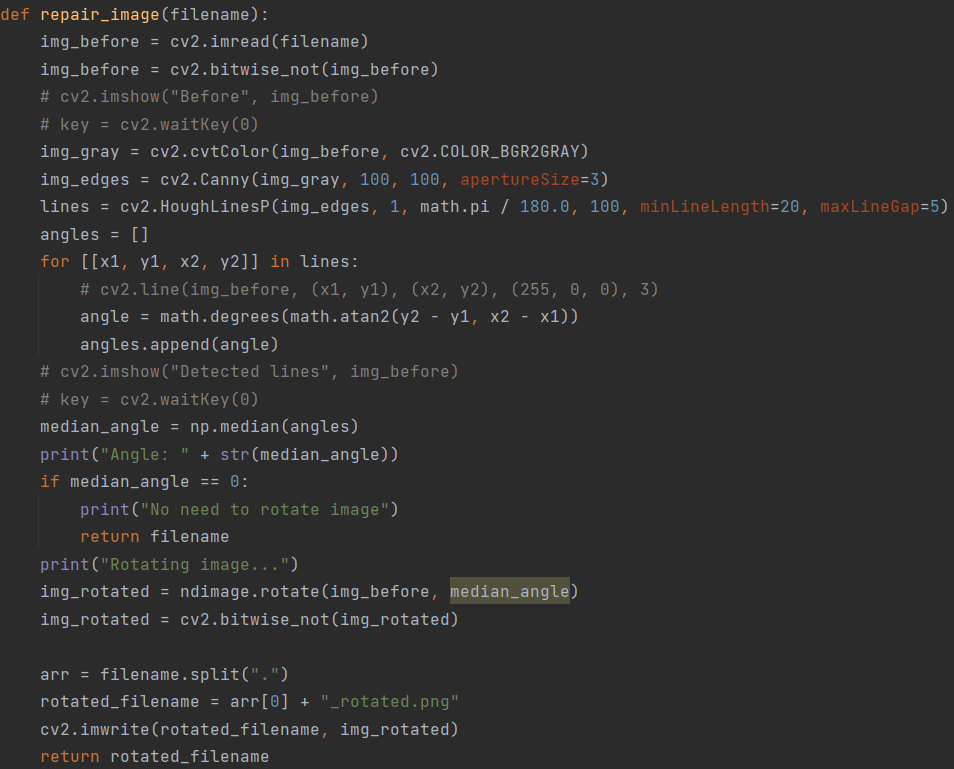


# Zadanie 2 – OCR

Napisz program przekształcający obraz w tekst, przyjmując następujące założenia:

1. Na obrazie znajduje się tekst złożony z małych liter alfabetu łacińskiego oraz cyfr.
2. Na obrazie występuje jeden typ i rozmiar czcionki.
3. Weź pod uwagę czcionki szeryfowe i bezszeryfowe.
4. W tekście występują znaki przestankowe: .,?!.
5. Tekst składa się z wielu linii.
6. Tekst może być obrócony (krzywo zeskanowany w całości).
7. Program powinien zwracać tekst wraz ze znakami białymi i znakami nowych linii.
8. Program może raportować procent poprawnie rozpoznanych liter dla pre-definiowanych obrazów testowych.
9. Program powinien dodatkowo zliczać liczbę wystąpień każdej litery.
10. Należy zastosować operacje splotu i DFT albo inne metody (klasyfikacja).
11. Należy dokonać redukcji szumu na obrazie wejściowym.

Na początek została napisana funkcja do obracania obrazu w razie potrzeby.



Następnie powstał początek głównej funkcji, która czyta wszystkie czcionki z folderu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przygotowujemy nasze dane z obrazu.

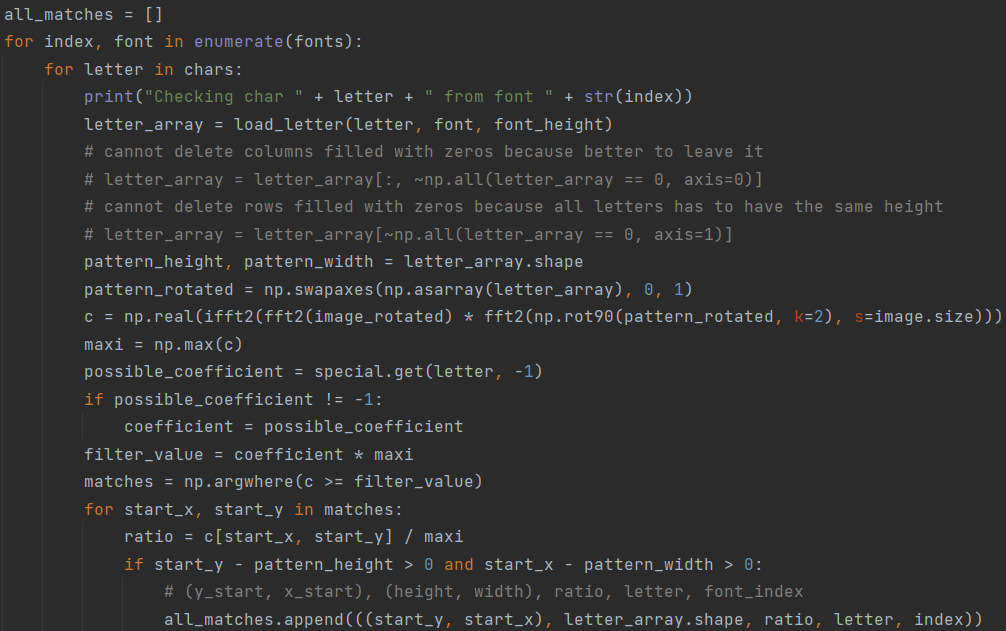


Funkcja ta ma też możliwość redukcji szumów.

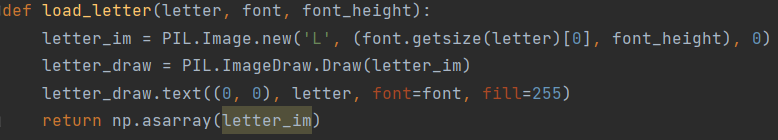
Obraz zawierający tekst

Opis wygenerowany automatycznie

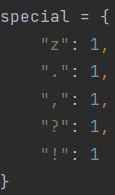
Wyszukujemy wszystkie pasujące znaki, z wcześniej zadeklarowanych.



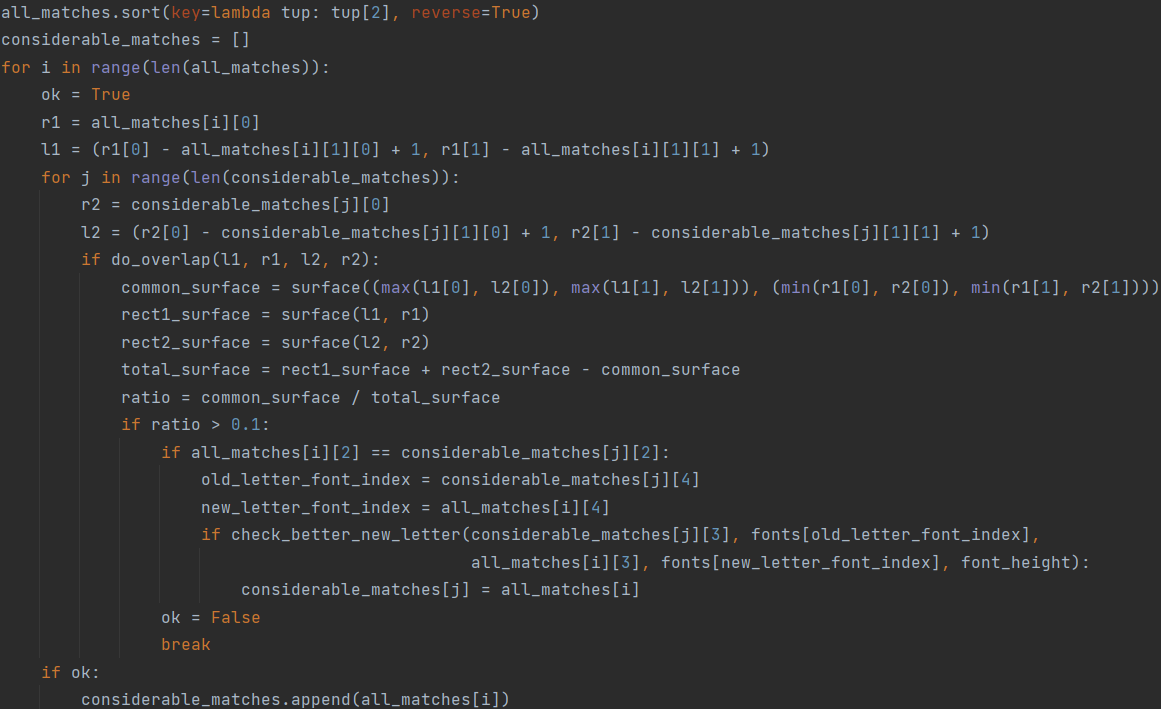
Ładowanie danej litery z określonej czcionki odbywa się przy pomocy poniższego kodu.



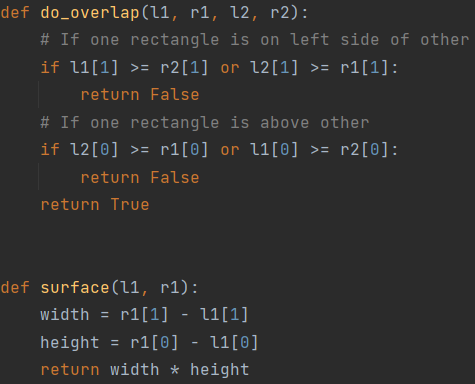
Dla niektórych liter zostały zadeklarowane specjalne warunki dopasowania.



Następnie zebrane dane są sortowane po współczynniku dopasowania.



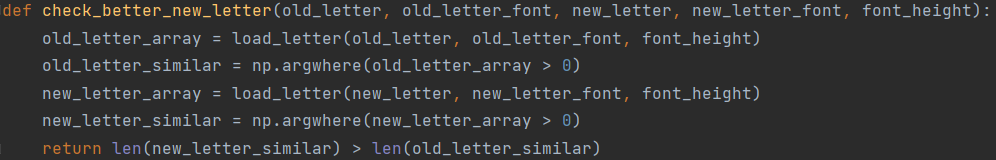
Znaki są sprawdzane między sobą, czy nie następują kolizje.



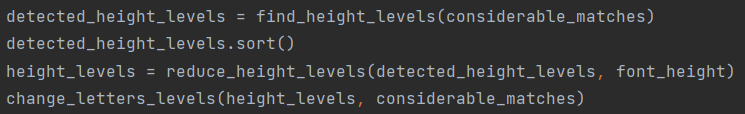
W przypadku kolizji są sprawdzane pewne warunki:

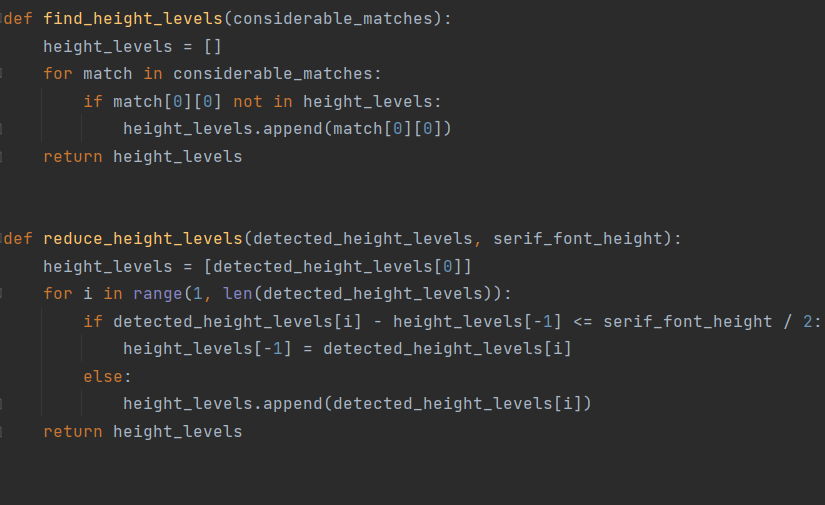
* Czy wspólna część („ratio”) jest większa od pewnej stałej („0.1”)?
* Czy współczynniki dopasowania są równe?
* Czy możliwy znak ma więcej pixeli w kolorze czarnym?

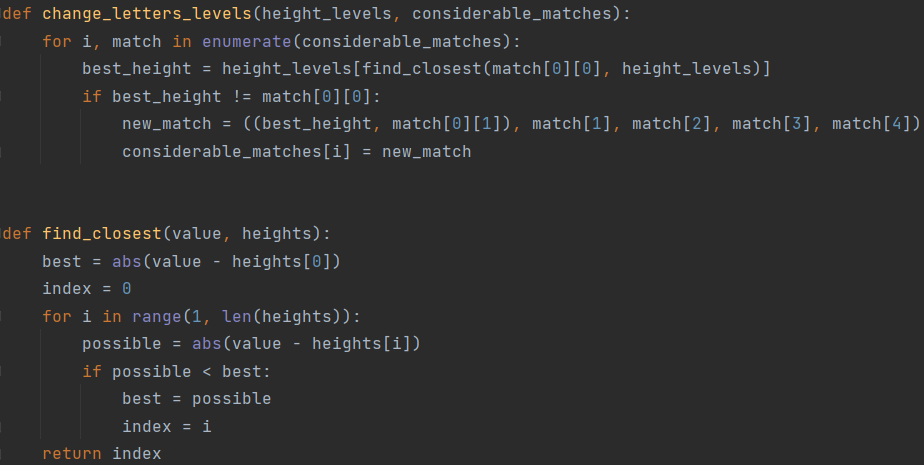
Jeśli on zajdą, to znak, który był wcześniej zostaje zastąpiony nowym.



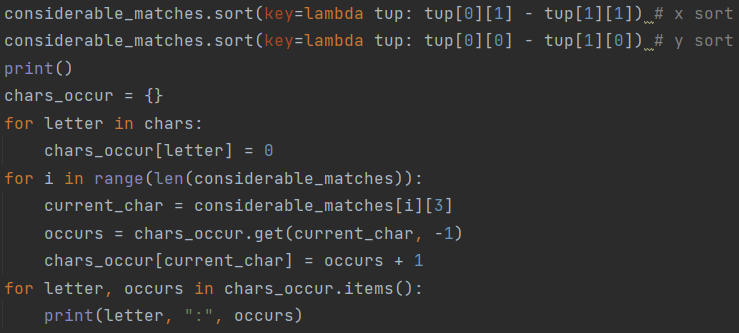
Następnie następuje redukcja podobnych poziomów wysokości, aby później łatwiej było odtworzyć oczekiwane rozwiązanie. Po wykonaniu poniższego kodu wszystkie znaki na podobnym poziomie będą mieć zmienioną wysokość na jedną z wartości, które zostały wcześniej znalezione.



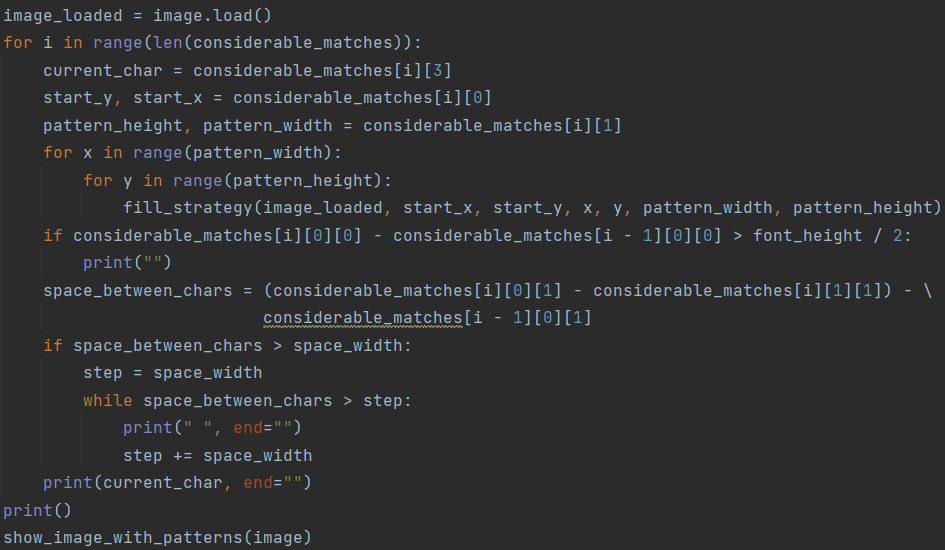


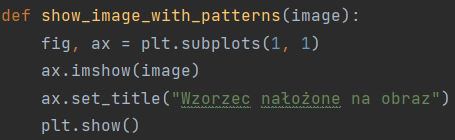


Następnie znaki są sortowane po współrzędnych „x”, a później po współrzędnych „y”. W kolejnym kroku zliczane są wystąpienia wszystkich rozpatrywanych liter.



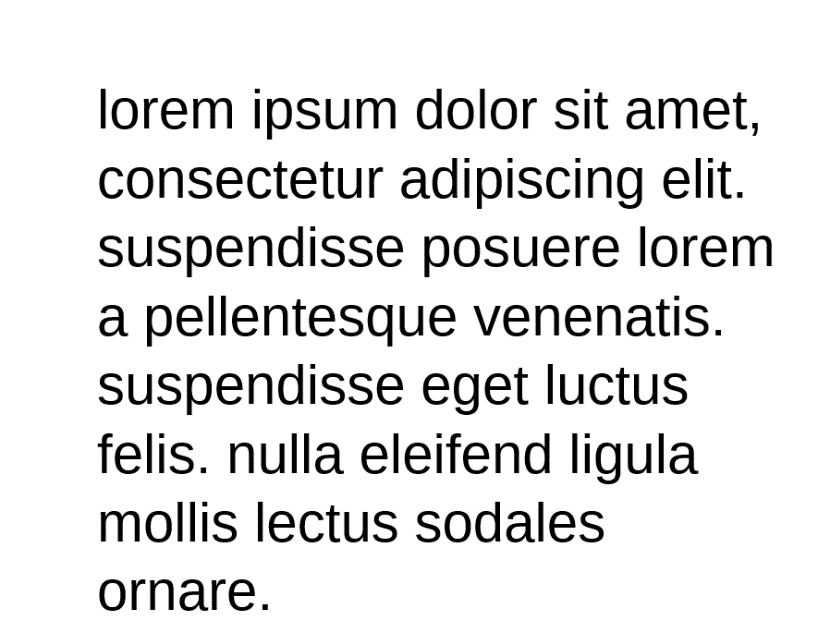
Na sam koniec tekst jest wypisywany do konsoli, a poszczególne znaki są zaznaczane na obrazie.





Zostało wykonanych kilka testów. **Nazleży zaznaczyć, że tekst był pisany w wordzie w dwóch czcionkach „Liberation Sans” oraz „Liberation Serif” w rozmiarze 36, a następnie były robione screeny tych tekstów. Współczynnik w testach był ustawiony na 0.94.**

Najpierw dla czcionki bezszeryfowej i bez obrotu.

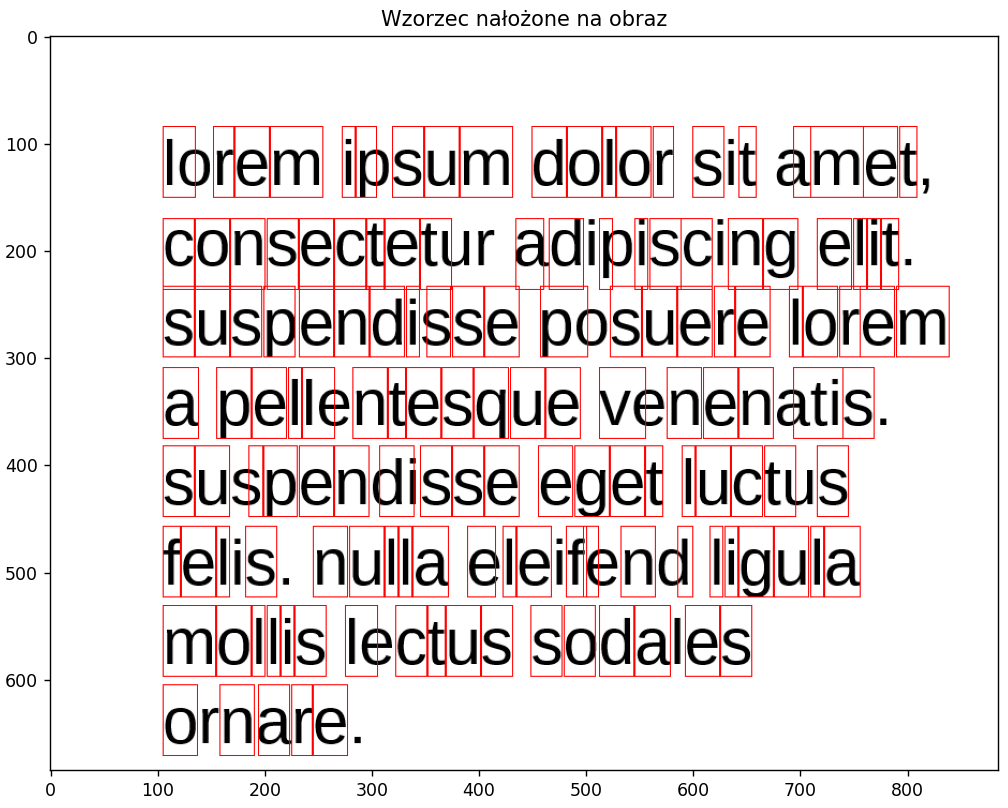


Wyniki były następujące.

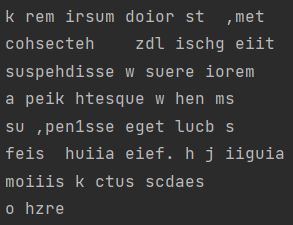
Dla tego tekstu nie trzeba było wykonać obrotu.



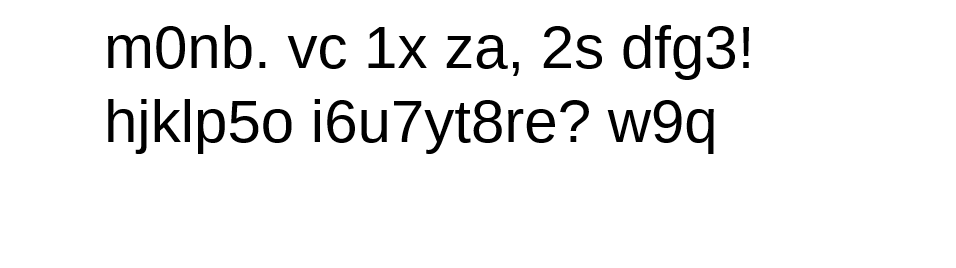
Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



A tekst uzyskany jest następujący.



Kolejny test czcionki bezszeryfowej i bez obrotu.

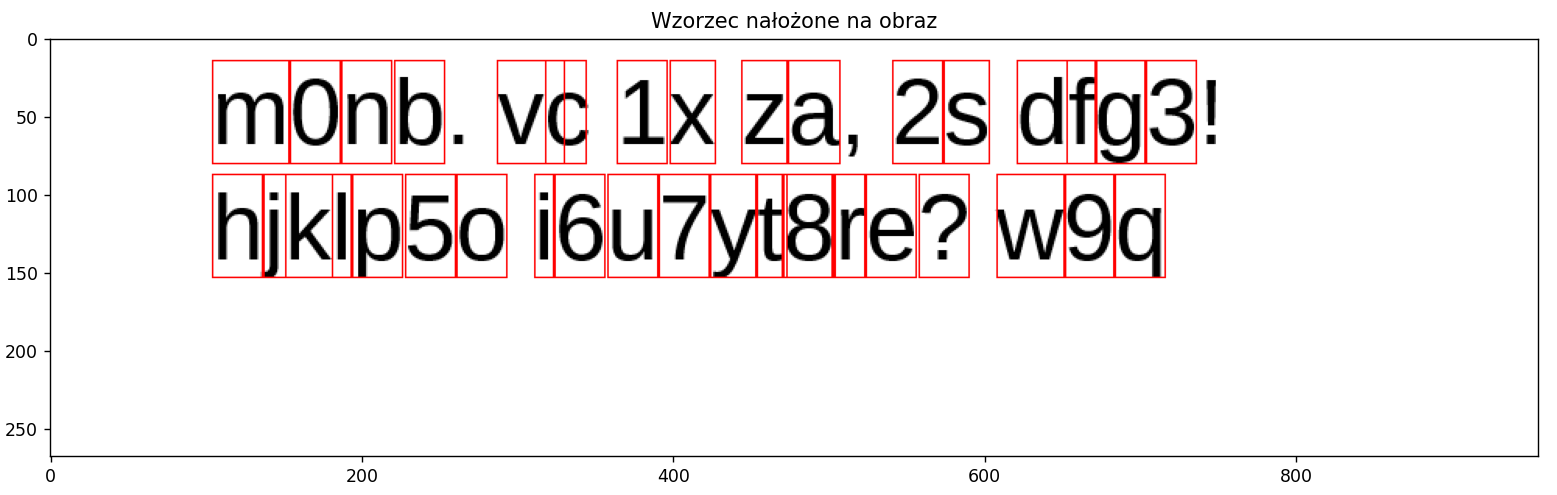


Wyniki były następujące.

Dla tego tekstu nie trzeba było wykonać obrotu.



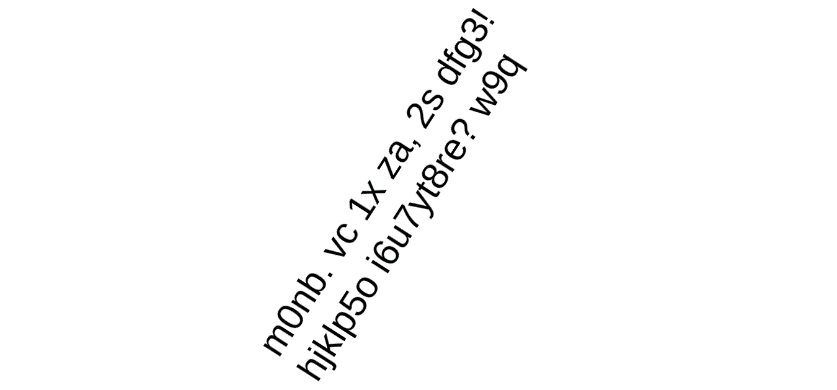
Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



A tekst uzyskany jest następujący.

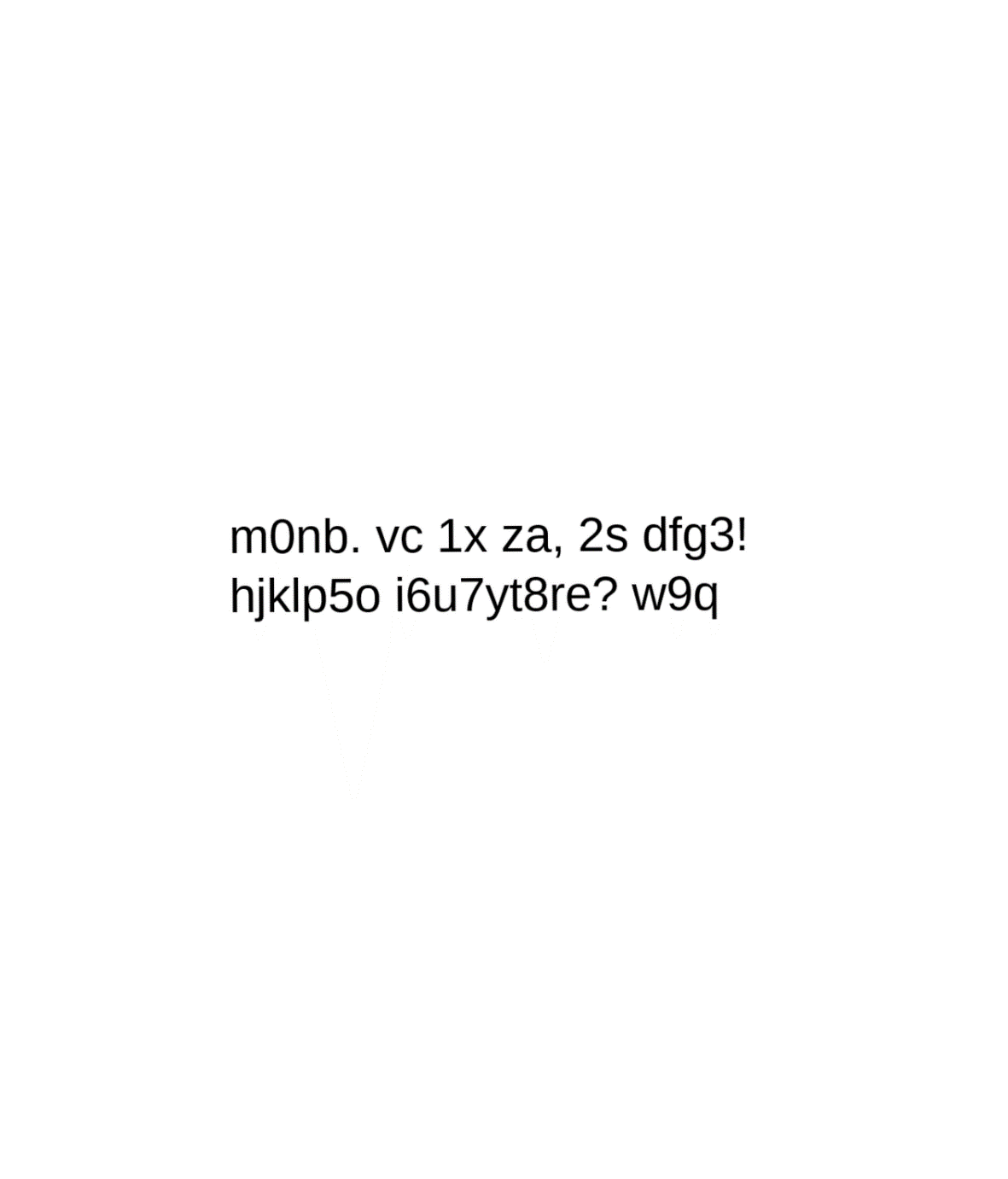


Teraz wykonano test czcionki bezszeryfowej z obrotem.



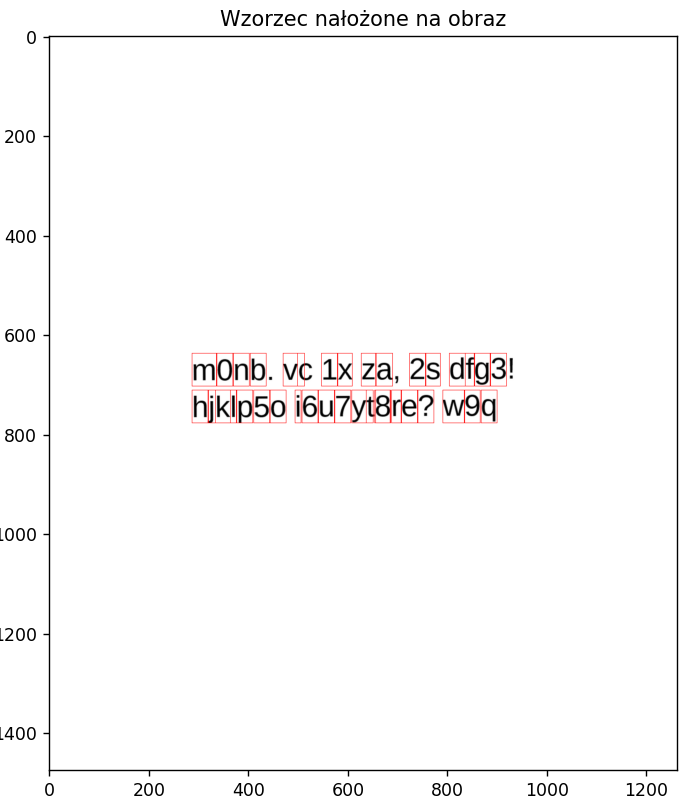
Program dokonał obrotu obrazu.





Wyniki były następujące.

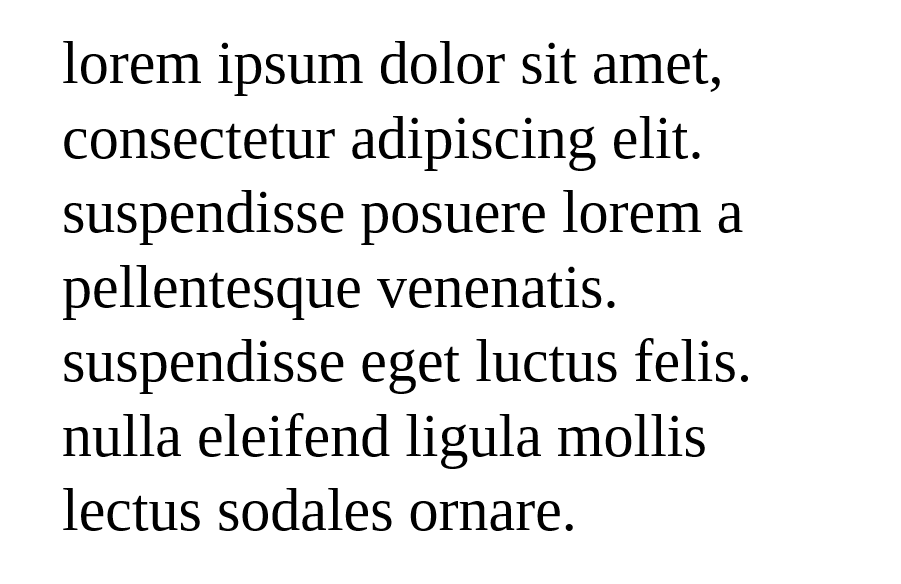
Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



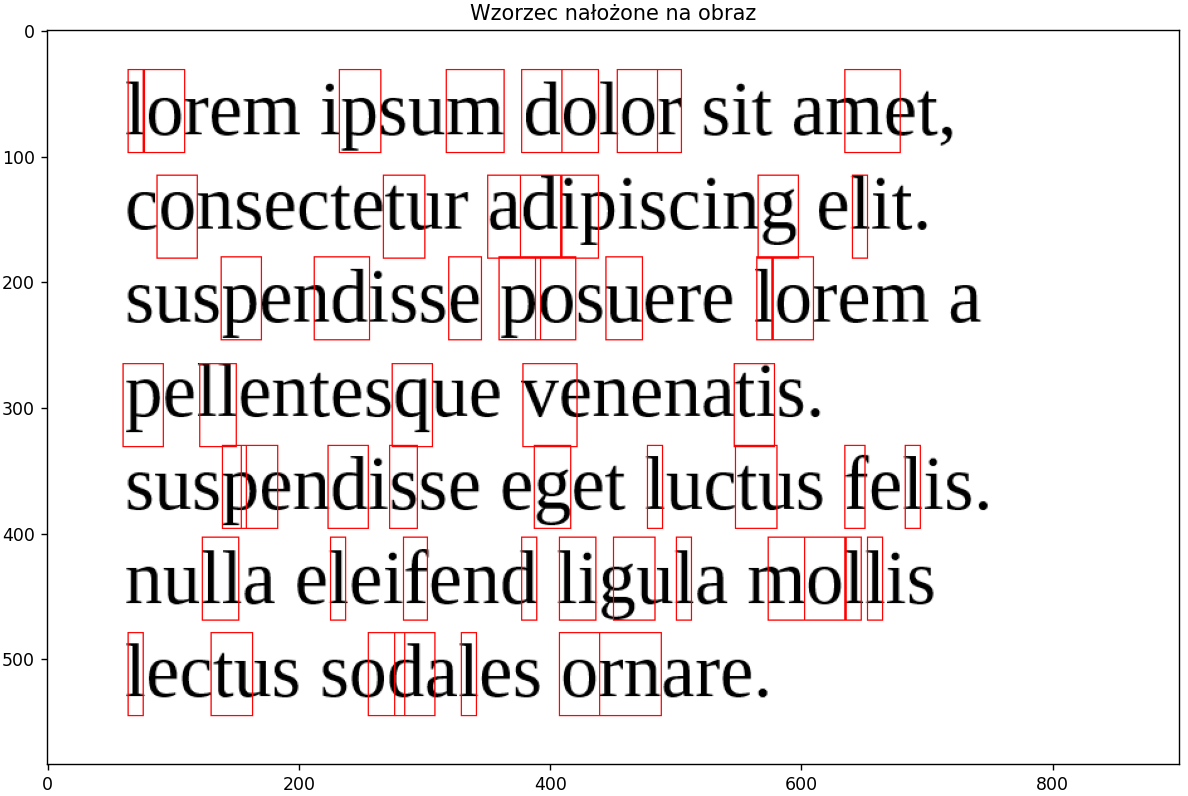
A tekst uzyskany jest następujący.



Podobne testy wykonano dla czcionki szeryfowej bez obrotu.



Wynik był strasznie marny.



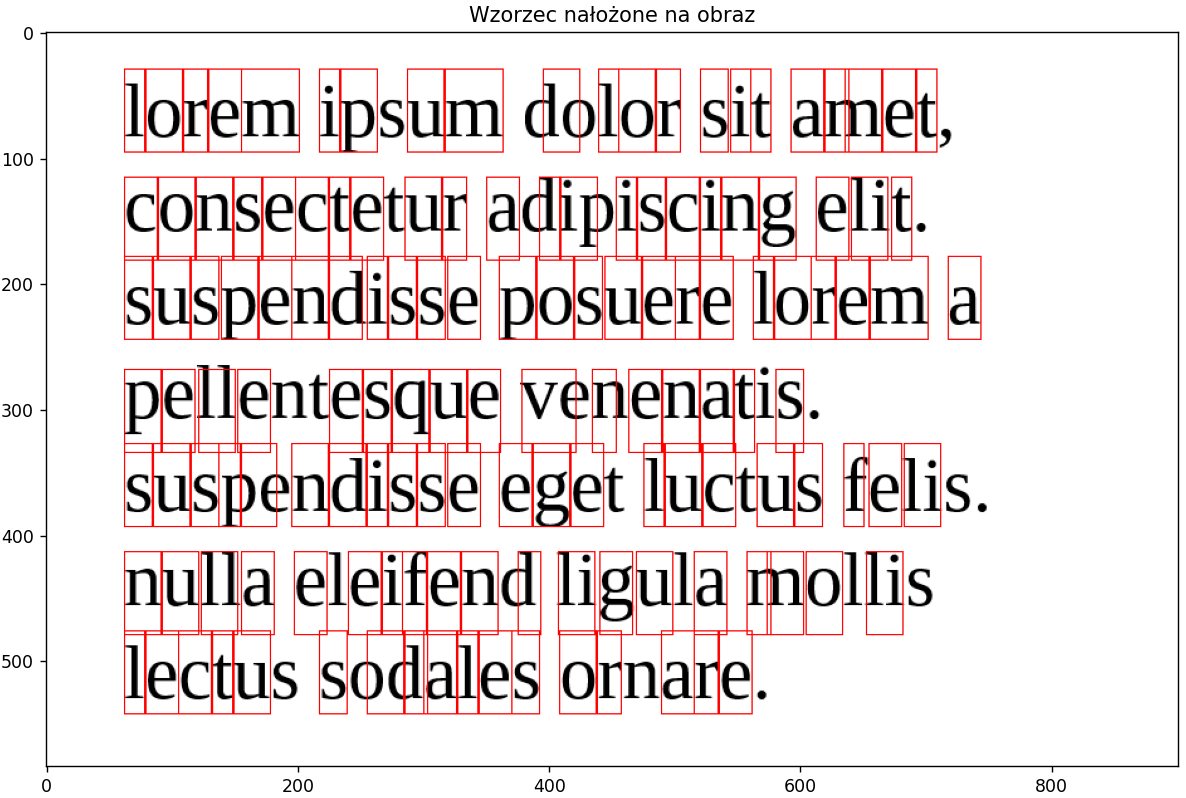
Dlatego program ten wywołano jedynie z czcionką szeryfową. Wynik był o wiele lepszy.

Wyniki były następujące.

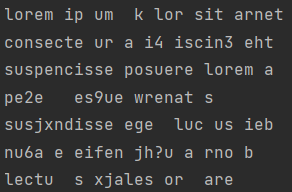
Dla tego tekstu nie trzeba było wykonać obrotu.



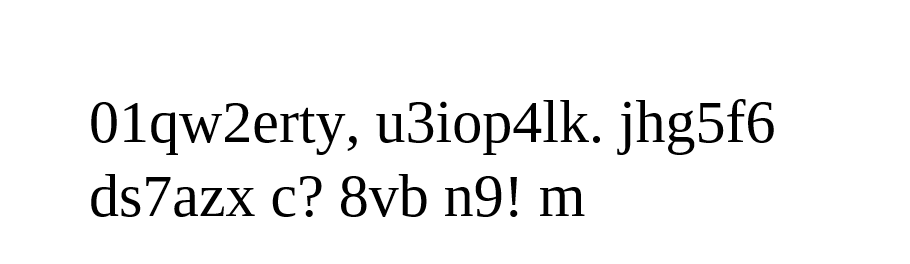
Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



A tekst uzyskany jest następujący.



Kolejny test dla czcionki szeryfowej bez obrotu.



Wyniki były następujące.

Dla tego tekstu nie trzeba było wykonać obrotu.



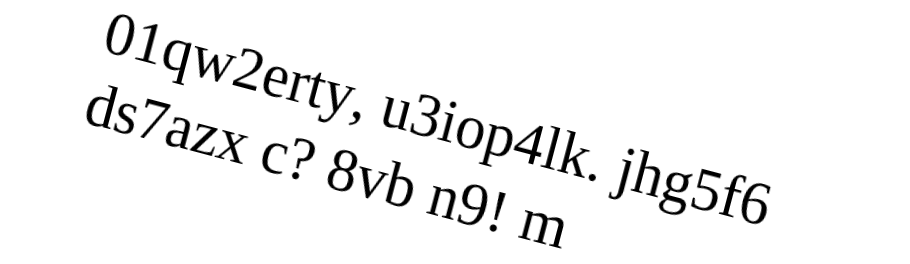
Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



A tekst uzyskany jest następujący.

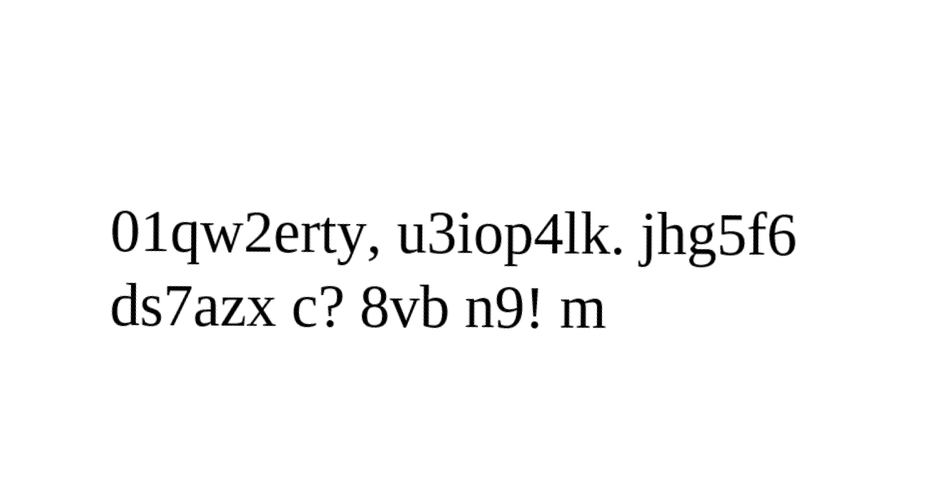


Teraz dokonano obrotu tekstu.



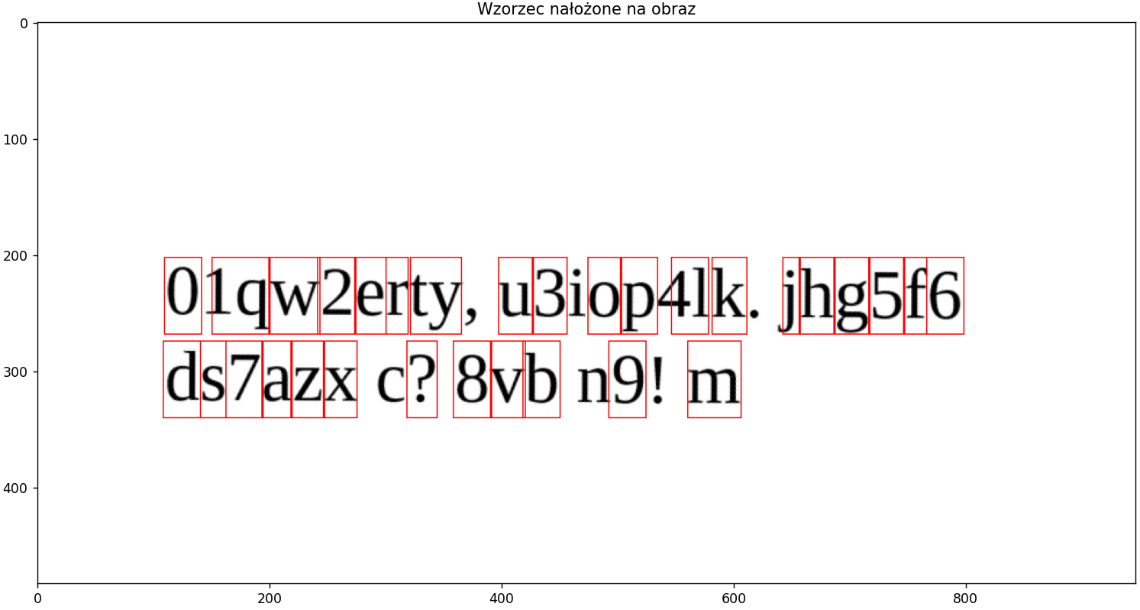
Program dokonał obrotu.





Wyniki były następujące.

Zaznaczone miejsca, w których znaleziono jakieś dopasowanie.



A tekst uzyskany jest następujący.

