POLITECHNIKA POZNAŃSKA
WYDZIAŁ ELEKTRYCZNY

Instytut Automatyki, Robotyki i Inżynierii Informatycznej

Bartosz Mila 131804
bartosz.mila@student.put.poznan.pl
Jakub Szczepaniak 131834
jakub.r.szczepaniak@student.put.poznan.pl
Paweł Świerczyński 131839
pawel.swierczynski@student.put.poznan.pl

Dokumentacja projektu

Wizualna identyfikacja komponentów elektronicznych

14 czerwca 2019

Spis treści

1 Uzasadnienie wyboru tematu

Temat wybrany został ze względu na wcześniejszą pracę z odczytywaniem tekstu z użyciem OCR (*Optical Character Recognition*). Dodatkowo temat wydawał się interesujący, gdyż łączy aspekty takie jak przetwarzanie obrazów, korzystanie z REST API, przygotowywanie interfejsu graficznego, obsługę plików JSON oraz wykorzystywanie algorytmów, takich jak odległość Levenshteina.

2 Podział prac pomiędzy członków zespołu

W rozdziale przedstawiono zadania wykonane przez poszczególnych członków zespołu.

I. Bartosz Mila:

   a. Prace nad CPP-REST i CMake.

   b. Program wykorzystujący Odległość Levenshteina.

   c. Utworzenie słownika komponentów.

   d. Dokumentacja.

II. Jakub Szczepaniak:

   a. Interfejs graficzny użytkownika.

   b. Obsługa silnika Tesseract.

   c. Funkcje związane z edycją obrazów.

   d. Dokumentacja.

III. Paweł Świerczyński:

   a. Obsługa Microsoft Computer Vision API.

   b. Obsługa plików JSON.

   c. Instalacja bibliotek.

   d. Dokumentacja.

3 Wymagania

Wymagania określają jakie funkcję ma spełniać system oraz sposób wykonywania określonych zadań. Wymagania można podzielić na:

3.1 Funkcjonalne

Wymagania funkcjonalne określają funkcjonalność systemu oferowaną przez aplikację.

I. Ogólne:

a. Wczytywanie obrazu w formacie JPG, PNG lub BMP. Z wczytanych obrazów można podjąć próbę odczytania tekstu.

b. Zapisywanie obrazu w formacie BMP.

c. Zapisywanie informacji o rozpoznanym komponencie w pliku tekstowym TXT. Informacje uzyskiwane są z przygotowanej bazy danych typu NoSQL zapisanej w formacie JSON.

d. Wybór sposobu rozpoznawania tekstu z obrazu między silnikiem Tesseract, a Microsoft Computer Vision API. Użytkownik może podjąć decyzję o tym, która z usług OCR zostanie wykorzystana do odczytywania tekstów z wczytanego obrazu. Do korzystania z Microsoft Computer Vision API wymagane jest połączenie z Internetem.

e. Rozpoznawanie tekstu z obrazu. Obraz może być rozpoznawany z użyciem silnika Tesseract lub Microsoft Computer Vision API.

f. Wyszukiwanie informacji o komponencie elektronicznym w bazie danych z użyciem odległości Levenshteina.

g. Zamknięcie aplikacji.

II. Edycja obrazów:

a. Dostosowywanie kontrastu obrazu.

b. Zamiana obrazu na obraz w skali szarości.

c. Zamiana obrazu na obraz czarno biały za pomocą progowania.

d. Obracanie obrazu o określony stopień.

e. Resetowanie modyfikacji wykonanych na obrazie.

III. Tesseract:

a. Wybór trybu pracy silnika Tesseract między Tesseract and Cube, Cube Only, Tesseract Only (Tryb Cube zużywa więcej zasobów i jest wolniejszy, ale daje lepsze rezultaty). Domyślnym trybem jest Tesseract and Cube. Teksty odczytywane są na podstawie danych przygotowanych dla języka angielskiego.

IV. Microsoft Computer Vision:

a. Wpisywanie klucza umożliwiającego korzystanie z API. Klucze mogą zostać wygenerowane na stronie produktu.

b. Wpisywanie URI serwera, do którego będzie się łączyła aplikacja. W przypadku korzystania z kluczy uzyskanych w 7-dniowym okresu próbnym domyślnym serwerem jest West Central US.

3.2 Pozafunkcjonalne

Wymagania pozafunkcjonalne określają właściwości i specyfikację programu.

I. Architektura:

a. Program napisany w języku C# z wykorzystaniem WPF.

b. Wykorzystanie biblioteki Tesseract do OCR.

c. Wykorzystanie REST API Microsoft Computer Vision do OCR.

d. Utworzenie słownika z komponentami elektronicznymi.

e. Obsługa plików JSON.

f. Zastosowanie bazy danych typu NoSQL.

II. Obrazy:

a. Możliwość wczytywania i zapisywania obrazów.

b. Możliwość edycji obrazu w celu poprawy wyników uzyskanych z OCR.

c. Możliwość podglądu edytowanych obrazów.

d. Obsługa plików o rozszerzeniach JPG, BMP i PNG.

III. Ogólne:

a. Program powinien być responsywny.

b. Do obsługi programu wymagana jest mysz komputerowa oraz klawiatura, natomiast minimalna rozdzielczość monitora wynosi 1280x720 px.

c. W celu korzystania z Microsoft Computer Vision potrzebne jest połączenie z Internetem.

d. Program dedykowany jest na komputery z zainstalowanym systemem operacyjnym Windows 10.

e. Interfejs programu napisany w języku angielskim.

4 Przypadki użycia

Przypadki użycia przedstawiają opis interakcji użytkownika z aplikacją.

UC01: Wczytywanie obrazu

Główny scenariusz:

1. Użytkownik chce wczytać obraz.
2. Użytkownik wybiera miejsce znajdowania się obrazu.
3. Użytkownik wczytuje obraz.

Rozszerzenia:

3.A. Użytkownik wybrał plik o rozszerzeniu innym niż BMP, PNG lub JPG.

3.A.1. System informuje o niepoprawnym rozszerzeniu pliku (powrót do 1.).

UC02: Zapisywanie obrazu

Główny scenariusz:

1. Użytkownik chce zapisać obraz.
2. Użytkownik wybiera miejsce ,w który ma zostać zapisany obraz.
3. Użytkownik zapisuje obraz.

Rozszerzenia:

3.A. Użytkownik nie podał rozszerzenia.

3.A.1. System zapisuje obraz z rozszerzeniem BMP.

3.B. Użytkownik podał rozszerzenie inne niż BMP.

3.B.1. System informuje o niepoprawnym rozszerzeniu pliku (powrót do 1.).

UC03: Zapisywanie informacji o rozpoznanym komponencie elektronicznym

Główny scenariusz:

1. Użytkownik chce zapisać informacje o rozpoznanym komponencie.
2. Użytkownik wybiera miejsce, w który mają zostać zapisane informacje.
3. Użytkownik zapisuje informacje.

Rozszerzenia:

3.A. Użytkownik nie podał rozszerzenia.

3.A.1. System zapisuje informacje w pliku z rozszerzeniem TXT.

3.B. Użytkownik podał rozszerzenie inne niż TXT.

3.B.1. System informuje o niepoprawnym rozszerzeniu pliku (powrót do 1.).

UC04: Resetowanie obrazu

Główny scenariusz:

1. Użytkownik chce zresetować obraz.
2. Użytkownik resetuje obraz.

Rozszerzenia:

2.A. Użytkownik zresetował obraz bez wczytanego obrazu.

2.A.1. System informuje o braku wczytanego obrazu (powrót do 1.).

UC05: Dostosowanie kontrastu

Główny scenariusz:

1. Użytkownik chce dostosować kontrast obrazu.
2. Użytkownik wybiera wartość kontrastu.
3. Użytkownik dostosowuje kontrast.

Rozszerzenia:

3.A. Użytkownik podał błędną wartość.

2.A.1. System wyświetla stosowną informację (powrót do 1.).

UC06: Zamiana obrazu na skalę szarości

Główny scenariusz:

1. Użytkownik chce zamienić obraz na skalę szarości.
2. Użytkownik zamienia obraz na skalę szarości.

Rozszerzenia:

2.A. Użytkownik nie wczytał obrazu.

2.A.1. System wyświetla informację o braku obrazu (powrót do 1.).

UC07: Binaryzacja obrazu

Główny scenariusz:

1. Użytkownik chce zamienić obraz na obraz czarno-biały.
2. Użytkownik podaje próg poniżej którego piksele zostaną zamienione na czarne.
3. Użytkownik dokonuje binaryzacji obrazu.

Rozszerzenia:

2.A. Użytkownik nie podał liczby z przedziały od 0 do 255.

2.A.1. System wyświetla informację o błędnej wartości (powrót do 2.).

3.A. Użytkownik nie wczytał obrazu.

3.A.1. System wyświetla informację o braku obrazu (powrót do 1.).

3.B. Użytkownik nie podał liczby z przedziały od 0 do 255.

3.B.1. System wyświetla informację o błędnej wartości (powrót do 2.).

UC08: Rotacja obrazu

Główny scenariusz:

1. Użytkownik chce obrócić obraz.
2. Użytkownik podaje stopień, o który ma się obrócić obraz.
3. Użytkownik obraca obraz.

Rozszerzenia:

3.A. Użytkownik nie wczytał obrazu.

3.A.1. System wyświetla informację o braku obrazu (powrót do 1.).

3.B. Użytkownik nie podał liczby całkowitej.

3.B.1. System wyświetla informację o błędnej wartości (powrót do 2.).

UC09: Odczytywanie tekstu z obrazu

Główny scenariusz:

1. Użytkownik chce odczytać tekst z obrazu.
2. Użytkownik wczytuje obraz, z którego chce odczytać tekst.
3. Użytkownik wybiera metodę, z pomocą której zostanie odczytany tekst z obrazu.
4. Użytkownik otrzymuje rozpoznany tekst.

Rozszerzenia:

4.A. Użytkownik wczytał obraz bez tekstu.

4.A.1. System informuje o braku znalezionego tekstu (powrót do 1.).

4.B. System nie znalazł lub nie odczytał tekstu.

4.B.1. System informuje o braku znalezionego tekstu (powrót do 1.).

UC10: Odszukiwanie informacji o komponencie elektronicznym

Główny scenariusz:

1. Użytkownik chce otrzymać informacje dotyczące komponentu elektronicznego.
2. Użytkownik podaje tekst odczytany z komponentu elektronicznego.
3. Użytkownik uruchamia wyszukiwanie informacji dotyczących danego elementu.
4. System przekazuje użytkownikowi informacje dotyczące komponentu elektronicznego.

Rozszerzenia:

2.A. Użytkownik nie podał tekstu odczytanego z komponentu elektronicznego.

2.A.1. System informuje o braku tekstu (powrót do 2.).

UC11: Wpisywanie klucza MS Computer Vision

Główny scenariusz:

1. Użytkownik chce wpisać klucz w aplikacji WIKE.
2. Użytkownik podaje klucz MS Computer Vision w odpowiednim miejscu aplikacji i zatwierdza go.

Rozszerzenia:

2.A. Użytkownik podał niepoprawny klucz.

2.A.1. Rozpoznanie tekstu z obrazu jest niemożliwe (powrót do 1.).

UC12: Wybieranie trybu działania silnika Tesseract

Główny scenariusz:

1. Użytkownik chce zmienić tryb działania silnika.
2. Użytkownik wybiera opcję zmiany działania silnika i interesujący go tryb.

UC13: Zmiana używanego serwera MS Computer Vision

Główny scenariusz:

1. Użytkownik chce zmienić używany serwer MS Computer Vision.
2. Użytkownik w odpowiednim miejscu aplikacji podaje link do strony serwera i zatwierdza go.

Rozszerzenia:

2.A. Użytkownik podał niepoprawny link.

2.A.1. Rozpoznanie tekstu z obrazu jest niemożliwe (powrót do 1.).

5 Wybrane technologie

W rozdziale przedstawiono zastosowane technologie. W przygotowaniu aplikacji wykorzystano następujące rozwiązania:

I. Narzędzia:

   a. Visual Studio 2017, Visual Studio 2019 - zintegrowane środowiska programistyczne umożliwiające tworzenie oprogramowania konsolowego oraz posiadające graficzny interfejs użytkownika,

   b. GitHub - serwis internetowy umożliwiający darmowy hosting programów - przeznaczony dla programistów,

   c. Windows Presentation Foundation (WPF) - zapewnia zunifikowany model programowania do tworzenia aplikacji desktopowych w systemie Windows.

   d. Microsoft Computer Vision API - Usługa przetwarzania obrazów platformy Azure oferuje deweloperom dostęp do zaawansowanych algorytmów przetwarzania obrazów i informacji zwrotnych. W celu analizy obrazu można przekazać obraz lub określić jego adres URL. Algorytmy przetwarzania obrazów pozwalają analizować zawartość na różne sposoby, w zależności od interesujących Cię funkcji wizualnych. Na przykład usługa przetwarzania obrazów może stwierdzić, czy na obrazie znajduje się zawartość przeznaczona dla dorosłych, lub znaleźć wszystkie twarze na obrazie. Przetwarzania obrazów w aplikacji można używać, korzystając z natywnego zestawu SDK lub wywołując interfejs API REST bezpośrednio.

II. Środowisko:

a. Visual Studio 2017,

b. Visual Studio 2019.

III. Biblioteki:

a. Tesseract 3.3.0 - jest to darmowy silnik do rozpoznawania znaków optycznych dla różnych systemów operacyjnych.

b. Json.NET 12.0.1 - popularna platforma ułatwiająca obsługę plików zapisanych w formacie JSON dla Microsoft.NET.

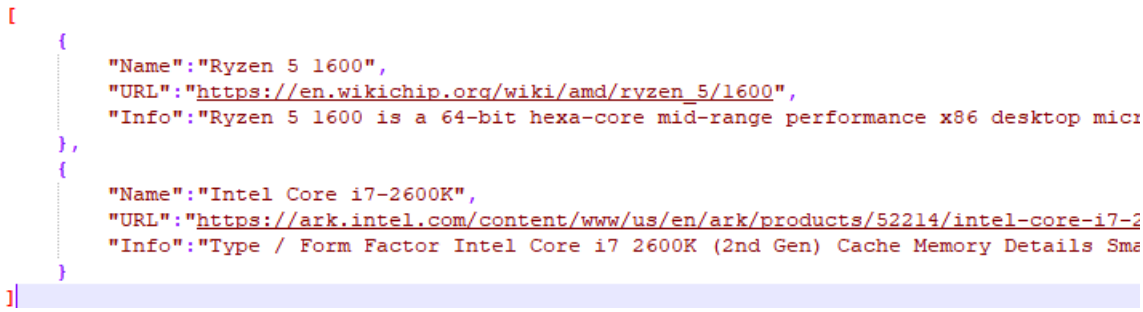
6 Schemat bazy danych

Ze względu na niską złożoność przechowywanych danych. Baza danych jest dokumentem w formacie JSON (Przykład rysunek 6.1) zawierającym informacje o komponentach elektronicznych. Plik JSON będzie zawierać następujące pola:

I. “Name” - nazwa komponentu elektronicznego,

II. “URL” - adres strony zawierającej informacje o komponencie,

III. “Info” - informacje o komponencie elektronicznym.



Rysunek 6.1: *Fragment pliku JSON*

7 Architektura rozwiązania

Program Wizualna Identyfikacja Komponentów Elektronicznych jest aplikacją desktopową, dedykowaną na komputery z systemem operacyjnym Windows. Aplikacja korzysta z dwóch bibliotek - Tesseract oraz Json.NET. Dodatkowo zaimplementowano w niej moduł służący do obsługi Microsoft Computer Vision API.

W programie wykorzystywana jest również baza danych typu NoSQL zawierająca informacje dotyczące komponentów elektronicznych. Ze względu na rodzaj przechowywanych informacji zdecydowano się na wybór formatu JSON. Komponenty elektroniczne przestawione w bazie danych wczytywane są podczas uruchamiania programu.

Po odczytaniu tekstu z wczytanego obrazu za określenie podobieństwa między tekstem rozpoznanym a tekstem identyfikującym dany komponent odpowiada dostosowany do sposobu działania aplikacji algorytm odległości Levenshteina. Stworzony w 1965 roku, algorytm ten służy do znajdowania miary odmienności wyrażeń tekstowych, poprzez obliczanie najmniejszej ilości działań prostych pozwalających na transformację jednego ciągu znaków w drugi. Za działania proste na napisach uznaje się podmianę, wstawienie lub usunięcie znaku.

Aplikacja łączy się z REST API udostępnianym przez Microsoft Computer Vision wykorzystując dwie metody:

- POST - służącą do przekazania danych obrazu do przetworzenia przez API wraz z dodatkowymi informacjami,
- GET - służącą do sprawdzenia stanu żądania oraz uzyskiwania wyników w postaci pliku JSON.

8 Napotkane problemy i ich rozwiązania

W początkowych fazach pracy nad aplikacją napotkano wiele problemów. Większość z nich wynikała z wyboru języka programowania C++ podłączenia bibliotek Tesseract i Leptonica do kompilatora. Nie znaleziono wystarczająco nowych skompilowanych bibliotek, a próby instalacji z wykorzystaniem vcpkg oraz cmake nie dały żadnych rezultatów. Rozwiązaniem tego problemu okazała się zmiana języka programowania na C#. Od tego momentu do instalacji silnika Tesseract wystarczyło wykorzystać menedżer pakietów NuGet.

Kolejnym napotkanym problemem było uzyskanie wyników z Microsoft Computer Vision API. W przypadku języka C++ podejmowaliśmy liczne próby znalezienia i wykorzystania biblioteki ułatwiającej korzystanie z REST API udostępnianego przez Microsoft Computer Vision. Ze względu na trudności w obsłudze bibliotek po zmianie języka programowania na C# skorzystaliśmy z proponowanego kodu udostępnionego w dokumentacji API.

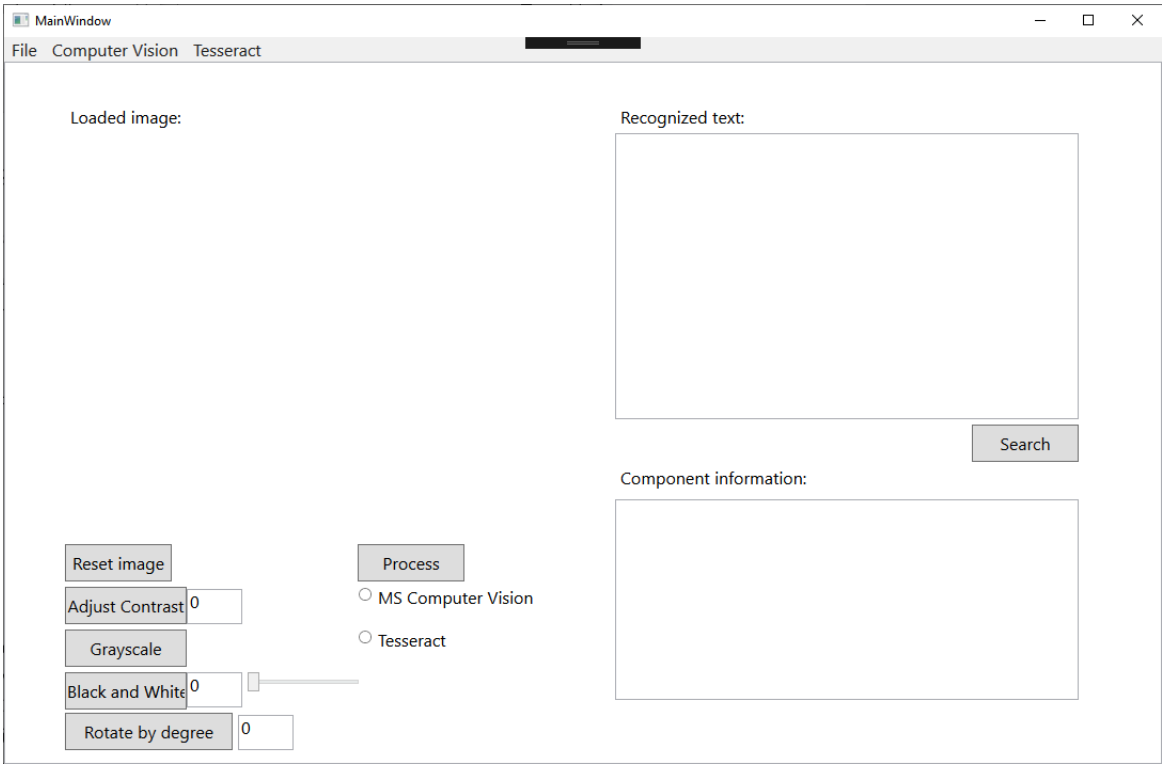
Ostatnim napotkanym problemem okazało się rozpoznanie tekstu z obrazów za pomocą silnika Tesseract. Tak jak API udostępniane przez Microsoft dawało bardzo dobre rezultaty to jednak nie jest one w pełni darmowe, posiada ograniczoną liczbę zapytań w miesiącu oraz wymaga dostępu do internetu. Aby uzyskać jakiekolwiek wyniki z użyciem darmowego Tesseracta obrazy musiały być w bardzo dobrej jakości z czarnym tekstem oraz w prawidłowej orientacji. Rozwiązaniem tego problemu są funkcje umożliwiające zwiększanie kontrastu, zamiany kolorów na skale szarości, progowanie oraz rotacja obrazów.

9 Instrukcja użytkowania aplikacji

W rozdziale przedstawiono szczegółową instrukcję użytkowania aplikacji.

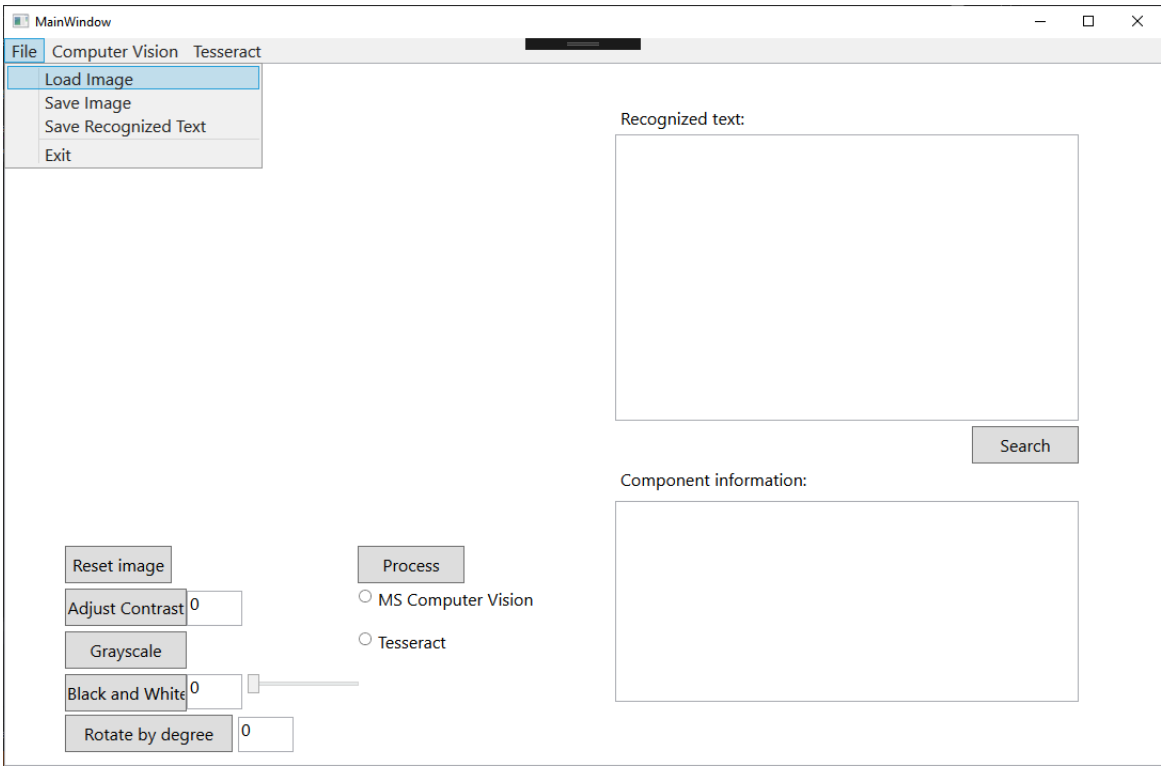
I. Podstawowe działanie

Aby korzystać z funkcjonalności aplikacji WIKE należy ją zainstalować na komputerze w wybranym katalogu. W momencie uruchomienia WIKE.exe pojawia się główne okno aplikacji, co ukazuje rysunek 9.1. Prawidłowa nawigacja pozwoli na wykorzystanie wszystkich funkcjonalności programu. Instrukcja wskaże użytkownikowi, w jaki sposób z niego korzystać.



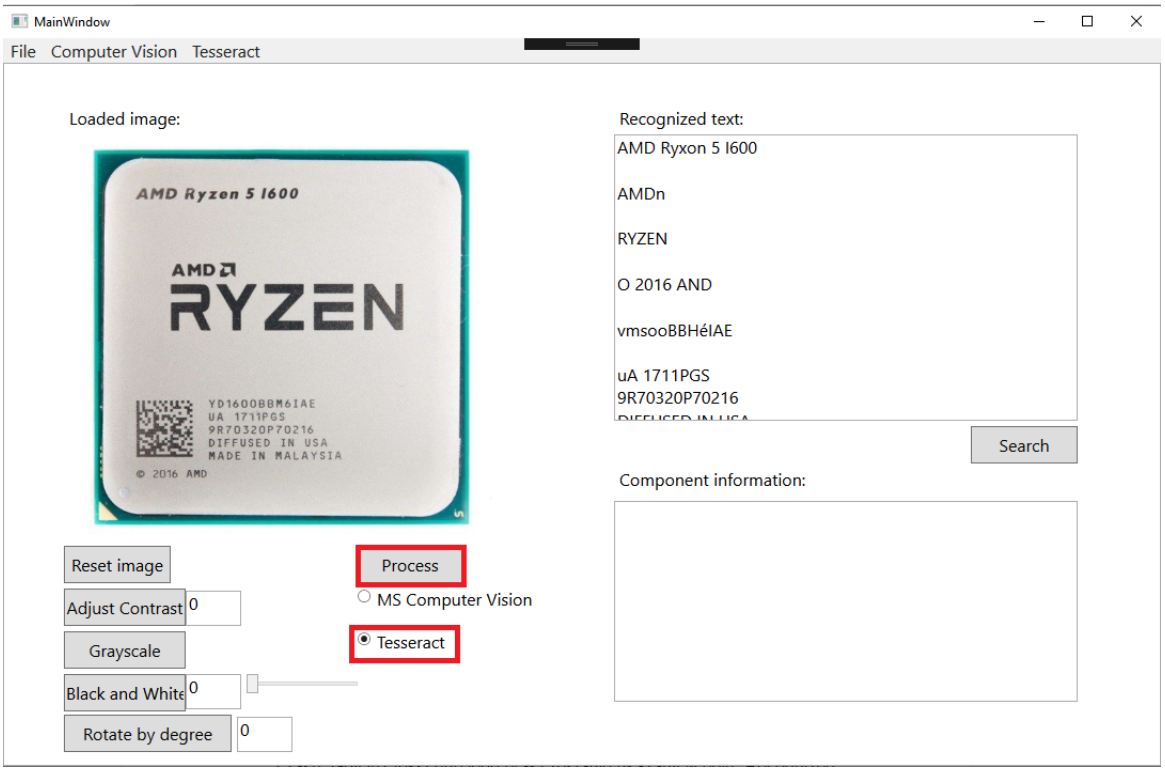
Rysunek 9.1: *Główne okno aplikacji*

Aby rozpocząć przetwarzanie obrazu, należy przejść do zakładki “File” i wybrać opcję “Load Image” (Rysunek 9.2). Następnie wymagane jest wybranie obrazu przedstawiającego komponent i znajdującego się na komputerze użytkownika (obraz powinien zawierać fragment tekstu jednoznacznie identyfikujący komponent). W tej samej zakładce znajdują się również opcje “Save Image” oraz “Save Recognized Text”, służące do zapisu efektów pracy z programem, a także “Exit” - kończąca działanie z programem. W zakładce “Tesseract” możliwy jest wybór sposobu działania tego trybu przetwarzania obrazów. Zastosowanie zakładki “Computer Vision” zostanie przedstawione w dalszej części instrukcji.



Rysunek 9.2: *Wybieranie obrazy*

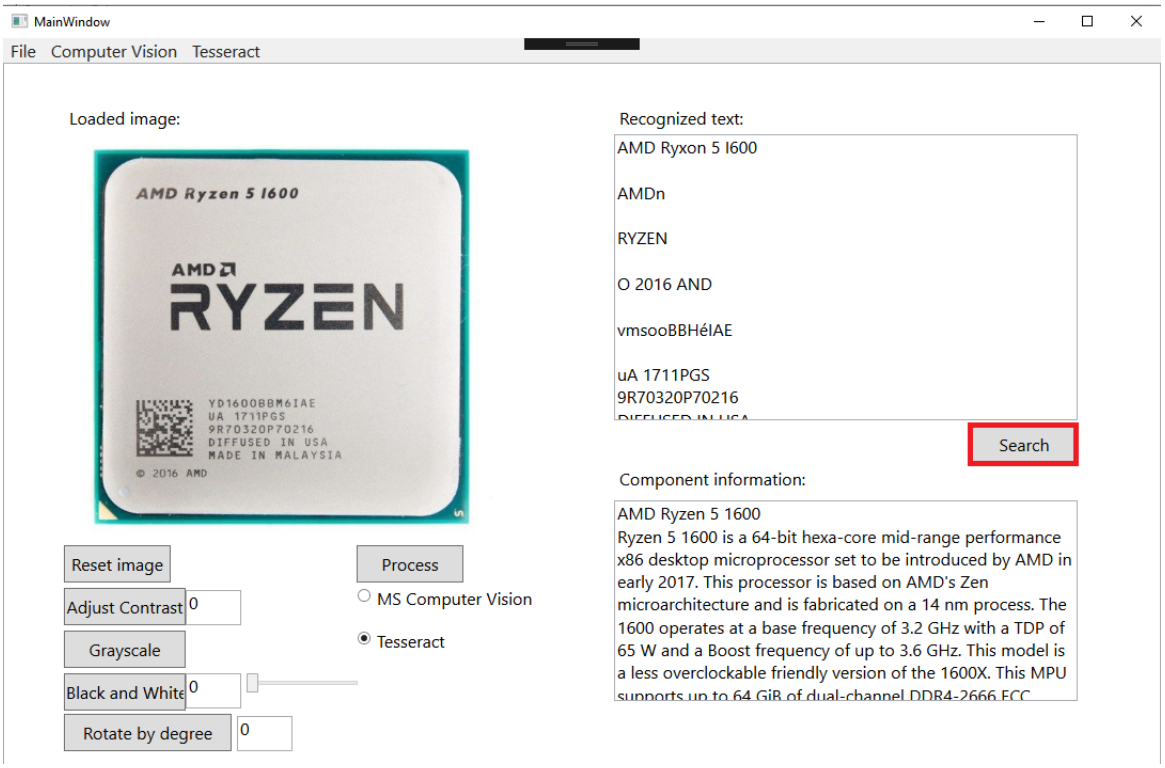
Po załadowaniu wybranego obrazu jego podgląd stanie się dostępny w oknie aplikacji. Gdy to nastąpi, możliwa jest próba odczytania tekstu na nim zawartego. Aby tego dokonać, należy wybrać sposób rozpoznawania tekstu (zaznaczyć pole Tesseract), a następnie przycisk "Process". Znalezione przez program litery, słowa i znaki pojawią się w polu "Recognized text" (Rysunek 9.3).



Rysunek 9.3: *Użycie OCR*

Prawidłowe wykonanie procedury rozpoznawania tekstu potrwa trochę czasu. Gdy już wykryty tekst (lub komunikat dotyczący jego braku) zostanie ukazany w polu "Recognized text", należy samodzielnie ocenić efekty. Jeśli są one zadowalające, należy spróbować wyszukać komponent w słowniku komponentów (przycisk "Search"), czego przykładowy

wynik ukazuje rysunek 9.4. Jeśli jednak wynik jest nieakceptowalny należy użyć funkcji służących do wprowadzania zmian w obrazie, co zostanie zaprezentowane w kolejnym fragmencie instrukcji.

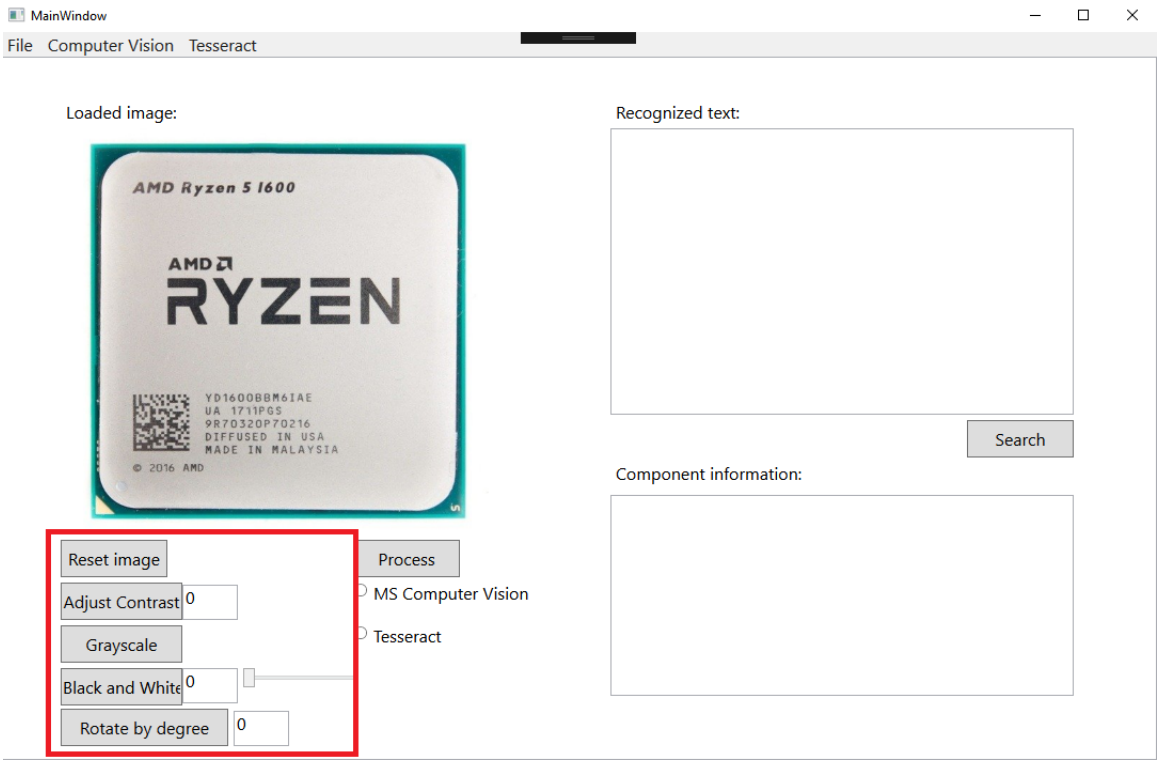


Rysunek 9.4: *Rozpoznanie komponentu*

II. Zmienianie parametrów obrazu

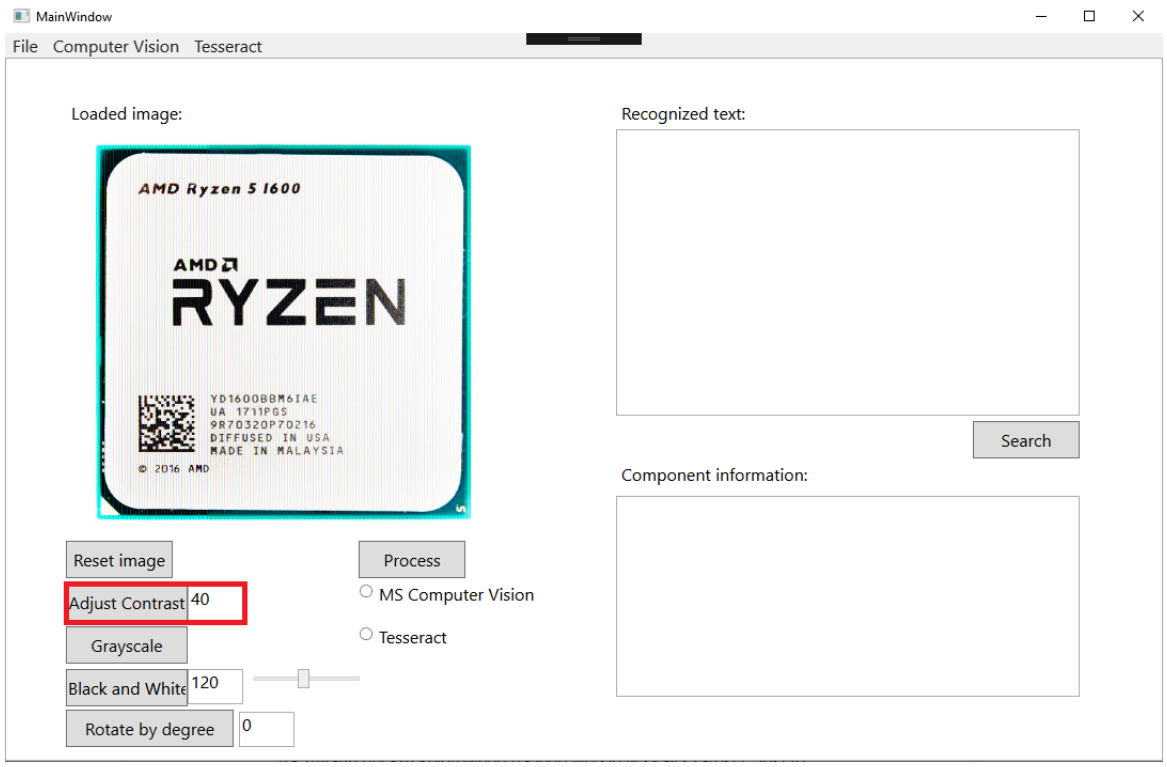
Możliwe jest, iż wynik przetwarzania obrazu będzie niewystarczający w celu rozpoznania komponentu. W takim przypadku rozwiązaniem problemu mogą okazać się odpowiednie funkcjonalności aplikacji, pozwalające na zmianę właściwości obrazu (Rysunek 9.5). Po satysfakcjonującej modyfikacji należy przetworzyć obraz ponownie. Odmienne funkcje są zalecane w przypadku odmiennych obrazów, co zostanie do-

kładnie omówione poniżej. Należy pamiętać, iż przycisk “Reset image” pozwala na przywrócenie obrazu do pierwotnej, niezmienionej postaci.



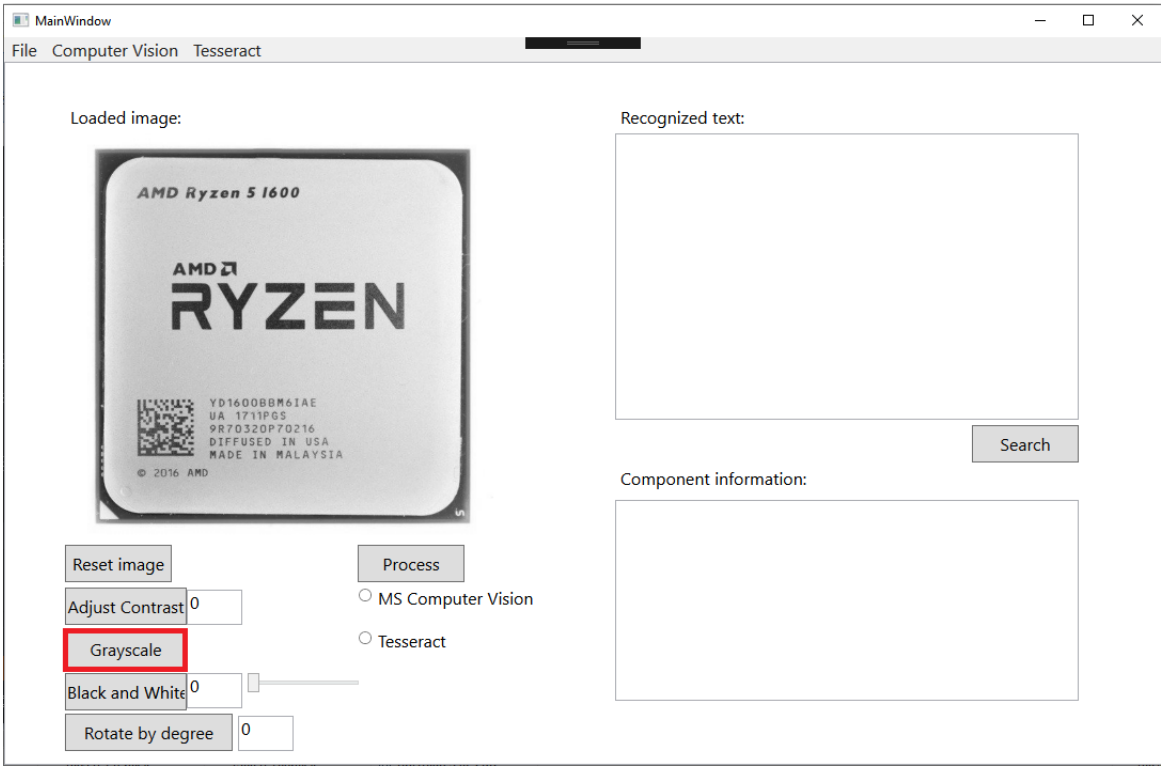
Rysunek 9.5: *Funkcje modyfikacji obrazu*

Pierwszą z dostępnych metod jest metoda “Adjust Contrast”. Pozwala ona na zmianę kontrastu obrazu o wartość ściśle zdefiniowaną przez użytkownika. Jest to przydatne w przypadku obrazów, w których napisy bądź oznaczenia komponentu są mało wyodrębnione od tła, na którym są umieszczone. Aby użyć tej funkcji należy podać wartość liczbową a następnie użyć przycisku “Adjust Contrast” (Rysunek 9.6).



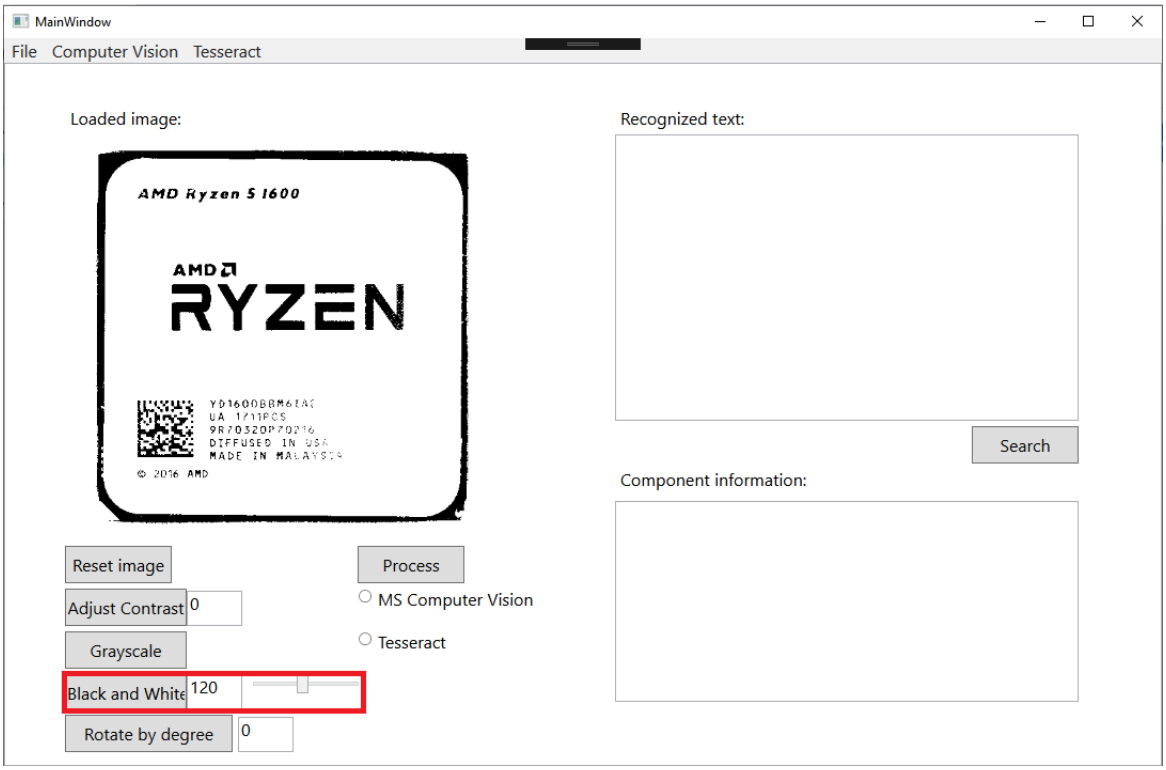
Rysunek 9.6: *Zwiększenie kontrastu obrazu*

Kolejną możliwością jest wykorzystanie opcji “Grayscale”. Pozwala ona na zmianę obrazu kolorowego na jego odwzorowanie w skali szarości. Jest to przydatne, gdy obraz jest bardzo kolorowy - gdy barwy w zauważalny sposób uniemożliwiają identyfikację tekstu. W tym przypadku nie ma możliwości podania żadnego parametru - użycie przycisku automatycznie zmieni obraz (Rysunek 9.7).



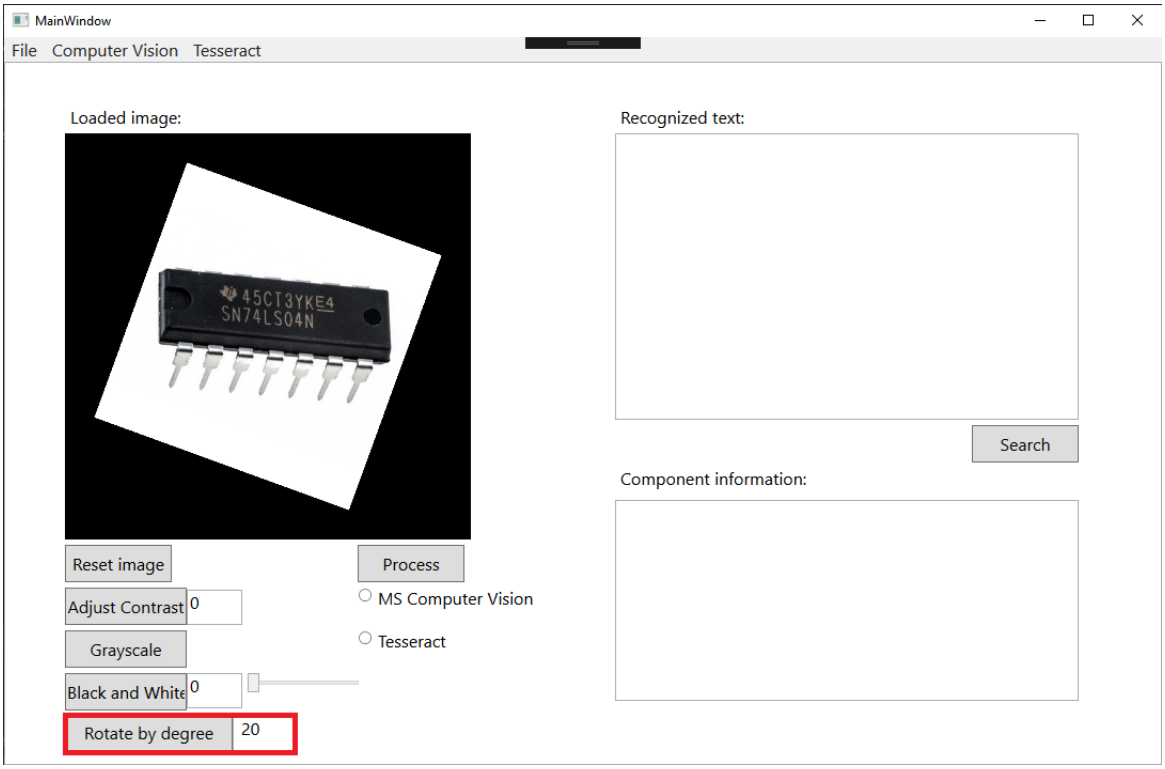
Rysunek 9.7: *Zmiana obrazu kolorowego na obraz w skali szarości*

Trzecią opcją zmiany parametrów obrazu jest “Black and White”. Funkcja ta jest progowaniem - podany obraz zostanie zamieniony na obraz monochromatyczny, zależny od progu ustawionego przez użytkownika. Opcja ta może okazać się pomocna, gdy na obrazie napisy będą niemal niewidoczne i kolorystycznie zbliżone do tła. Odpowiedni próg pozwoli na ich wyodrębnienie. Aby wykonać progowanie należy po ustaleniu jego wartości (za pomocą wyznaczonego pola bądź suwaka) użyć przycisku “Black and White” (Rysunek 9.8).



Rysunek 9.8: *Użycie funkcji progowania*

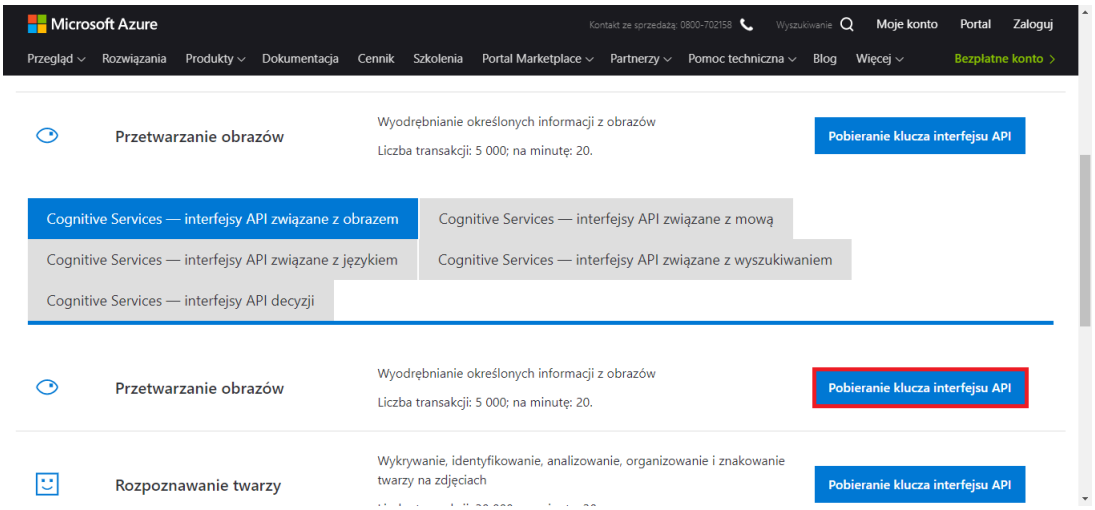
Ostatnią metodą zmiany parametrów obrazu jest "Rotate by degree". Jest to obrócenie obrazu o podany przez użytkownika kąt. Ta opcja jest użyteczna w przypadku obrazów, na których tekst identyfikujący komponent jest na tyle przekrzywiony, by utrudniało to jego odczytanie. Aby skorzystać z tej opcji należy wpisać ilość stopni, o jaką obraz ma zostać obrócony (w prawo), a następnie użyć przycisku "Rotate by degree" (Rysunek 9.9).



Rysunek 9.9: *Obracanie obrazu*

III. Microsoft Computer Vision

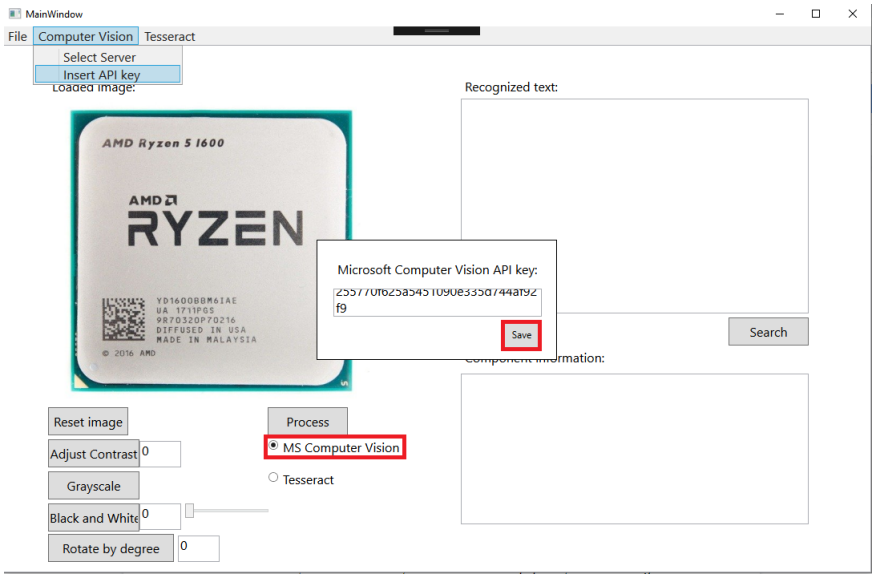
Alternatywnym i znacznie dokładniejszym sposobem przetwarzania obrazów jest wykorzystanie narzędzia Microsoft Computer Vision. Aby skorzystać z tej opcji, konieczne jest posiadanie aktywnego klucza dostępu do tej usługi. Aby go uzyskać, można np. przejść do strony: https://azure.microsoft.com/pl-pl/try/cognitive-services/?api=computer-vision. Gdy strona zostanie otworzona, należy wybrać "Pobieranie klucza interfejsu API" tak, jak to ukazano na rysunku 9.10.



Rysunek 9.10: *Uzyskanie klucza dostępu MS Computer Vision*

Otworzone zostanie okno w którym należy wybrać typ klucza i sposób jego pozyskania. Niniejsza instrukcja będzie dotyczyć uzyskania bezpłatnego, 7-dniowego klucza dostępu. Po dokonaniu takiego właśnie wyboru konieczny jest wybór regionu i zalogowanie przez osobiste konto Microsoft, Facebook, LinkedIn lub GitHub.

Gdy klucz zostanie uzyskany należy w głównym oknie aplikacji WIKE wybrać zakładkę "Computer Vision", a następnie opcję "Insert API key" (w tej zakładce możliwe jest również wybranie serwera MS Computer Vision). W kolejnym kroku wpisać uzyskany klucz, zatwierdzić i zaznaczyć "MS Computer Vision" jako sposób przetwarzania obrazu (Rysunek 9.11). Dalsze przetwarzanie jest analogiczne jak w przypadku opcji "Tesseract".



Rysunek 9.11: *Wykorzystanie MS Computer Vision*

10 Bibliografia

[1] Microsoft Computer Vision, https://azure.microsoft.com/pl-pl/services/cognitive-services/computer-vision/ (dostęp: 14.06.2019)

[2] Odległość Levenshteina, https://en.wikipedia.org/wiki/Levenshtein_distance (dostęp: 14.06.2019)

[3] Tesseract OCR, https://github.com/tesseract-ocr/tesseract (dostęp: 14.06.2019)