

**UNIWERSYTET JANA DŁUGOSZA w CZĘSTOCHOWIE**



**Wydział Nauk Ścisłych, Przyrodniczych i Technicznych**

Kierunek: Informatyka

Specjalność: Inżynieria oprogramowania

**Grzegorz Gromko**

nr albumu: 80385

**Projekt i implementacja programu do optycznego  
rozpoznawania i przetwarzania zapisu nutowego do  
formatu MIDI w języku Python**

**Design and implementation of a program for optical  
recognition and processing of musical notation to MIDI  
format in Python**

Praca inżynierska przygotowana  
pod kierunkiem  
dr hab. Andrzej Zbrzezny, prof. UJD

Częstochowa, 2024

# Spis treści

<b>Wstęp</b> . . . . .	2
<b>1. Specyfikacja wymagań</b> . . . . .	3
1.1. Wymagania biznesowe . . . . .	3
1.2. Wymagania funkcjonalne . . . . .	3
1.3. Wymagania нефункционалне . . . . .	3
<b>2. Technologie wykorzystane w projekcie</b> . . . . .	4
2.1. Python 3 . . . . .	4
2.2. PyTorch . . . . .	4
2.3. OpenCV . . . . .	5
2.4. Pillow . . . . .	5
2.5. Verovio Toolkit . . . . .	5
<b>3. Budowa programu</b> . . . . .	6
3.1. Schemat struktury programu . . . . .	6
<b>4. Użyte narzędzia programistyczne</b> . . . . .	7
4.1. Git . . . . .	7
4.2. Visual Studio Code . . . . .	7
4.3. Google Colab . . . . .	7
4.4. title . . . . .	7
<b>5. Implementacja programu</b> . . . . .	8
5.1. Podrozdział . . . . .	8
<b>Zakończenie</b> . . . . .	9
<b>Bibliografia</b> . . . . .	10
<b>Spis tabel</b> . . . . .	11
<b>Spis rysunków</b> . . . . .	12

## **Streszczenie**

Celem niniejszej pracy jest stworzenie programu służącego do optycznego rozpoznawania zapisu nutowego z plików graficznych oraz przetworzenia wydobytych informacji semantycznych do pliku w formacie MIDI. Program posiada system automatycznego usuwania wypaczania obrazu, a także możliwość utworzenia pliku graficznego z uzyskanych danych, pozwalając na digitalizację zapisu nutowego.

# **Wstęp**

Tu wprowadzenie w tematykę, motywacja, cel pracy, układ rozdziałów.

## **ROZDZIAŁ 1**

# **Specyfikacja wymagań**

W tym rozdziale zawarto wymagania postawione programowi przeznaczonemu do odczytywania notacji muzycznej z plików graficznych oraz konwersją odczytanych danych do formatu MIDI.

### **1.1. Wymagania biznesowe**

Nazwa programu to "Photo-2-MIDI". Przeznaczeniem programu jest umożliwienie użytkownikom zdigitalizowanie zapisów nutowych ze zdjęć lub skanów nut do plików MIDI.

### **1.2. Wymagania funkcjonalne**

1. Użytkownik może wprowadzić

### **1.3. Wymagania нефunkcjonalne**

1. Program wymaga do działania zainstalowanego języka Python w wersji 3.11

## ROZDZIAŁ 2

# Technologie wykorzystane w projekcie

W drugim rozdziale przedstawiono technologie wykorzystane w implementacji programu. Głównymi czynnikami wyboru danych technologii była łatwość użycia oraz ich możliwości.

Najważniejszymi punktami wybranych technologii są m.in.:

- Otwarty kod źródłowy
- Popularność
- Ogólnodostępność
- Aktywna społeczność rozwijająca projekty
- Łatwość wykorzystania
- Dostępności informacji dot. technologii

Dzięki powyższym cechom wykazywanym przez użyte technologie mamy zapewnioną szybkość tworzenia oprogramowania, gdzie w przypadku napotkania błędów w procesie jego kreacji możemy liczyć na pomoc społeczności w jego rozwiązaniu jak i szeroką gamę wcześniej stworzonych narzędzi.

## 2.1. Python 3

Sercem każdego programu jest język, w którym został on napisany. Językiem programowania, wybranym do implementacji projektu jest **Python** w wersji 3.11. Jest to wysokopoziomowy, wieloparadygmataowy język skryptowy ogólnego przeznaczenia, którego ideą przewodnią jest czytelność uzyskanego kodu źródłowego. Powstał w 1991 roku i jest ciągle rozwijany oraz cieszy się od wielu lat niesłabnącą popularnością, która wynika między innymi z:

- Łatwość nauki
- Dostępności materiałów dydaktycznych
- Wszechstronności języka
- Mnogości różnorodnych bibliotek
- Wieloplatformowość

## 2.2. PyTorch

Otwartoźródłowa biblioteka uczenia maszynowego stworzona przez Facebook AI Research (teraz Meta AI), bazowana na Pythonie i Torch-u, głównie używana do aplikacji uczenia maszynowego używających procesorów graficznych i TPU. PyTorch jest preferowany przez wielu badaczy sztucznej inteligencji przez używanie dynamicznych grafów obliczeń, co pozwala na uruchamianie i testowanie małych części kodu w czasie rzeczywistym bez potrzeby implementacji całości kodu, móc sprawdzić czy dana część kodu działa czy nie.

Głównymi zaletami PyTorch są:

- Obliczenia tensorowe z dobrym wsparciem przyspieszenia sprzętowego GPU i TPU
- Automatyczne różnicowanie dla tworzenia i trenowania głębokich sieci neuronowych

## 2.3. OpenCV

OpenCV (Open Source Computer Vision Library) jest otwartoźródłową biblioteką widzenia maszynowego oraz uczenia maszynowego. Została ona zbudowana by dostarczyć wspólną infrastrukturę dla aplikacji rozpoznawania obrazów oraz przyspieszenia użycia percepcji maszynowej w produktach komercyjnych.

## 2.4. Pillow

Biblioteka będąca odgałęzieniem Python Imaging Library (PIL), która to dodaje możliwości przetwarzania obrazów do programów pisanych w języku Python. Dodaje ona szerokie wsparcie formatów plików, wydajną wewnętrzną reprezentację oraz duże możliwości przetwarzania obrazów.

## 2.5. Verovio Toolkit

Verovio jest szybką, przenośną i lekką biblioteką do grawerowania cyfrowych zapisów nutowych MEI (Musical Encoding Initiative) do obrazów formatu SVG. Zawiera również konwertery do renderowania cyfrowych zapisów nutowych formatów MusicXML, Musedata, EsAC czy Humdrum.

## **ROZDZIAŁ 3**

# **Budowa programu**

W rozdziale trzecim zawarto informacje dotyczące budowy programu. Na początku opisana jest struktura programu w postaci diagramów. Następnie opisane są poszczególne fragmenty programu. Na końcu znajduje się opis interfejsu graficznego programu.

### **3.1. Schemat struktury programu**

...



## ROZDZIAŁ 4

# Użyte narzędzia programistyczne

Rozdział ten zawiera opis narzędzi programistycznych wykorzystywanych podczas pracy nad projektem.

### 4.1. Git

Jedno z najbardziej powszechnych i podstawowych narzędzi programistycznych, umożliwiające rozproszoną kontrolę wersji plików, która w przeciwieństwie do wielu alternatyw odbywa się lokalnie, gdyż każda kopia plików źródłowych kontrolowanych przez Gita może być pełnoprawnym repozytorium zawierającym całą historię zmian zachodzących na tychże plikach. Pozwala to na powrót do konkretnych wersji, których programista może potrzebować. Narzędzie to jest szeroko wykorzystywane do rozproszonego tworzenia oprogramowania przez grupy programistów w ramach jednego projektu, gdzie zmiany wprowadzane przez różnych ludzi są scalane do pojedynczego źródła, wykorzystując komercyjne rozwiązania hostowania online do przechowywania historii zmian.

Głównymi założeniami Gita jest:

- Szybkość działania
- Elastyczność
- Bezpieczeństwo
- Wsparcie nielinowych, rozproszonych przepływów pracy
- Integralność danych

### 4.2. Visual Studio Code

Darmowy, otwartoźródłowy program do edycji, analizy i zarządzania kodem, stworzony i rozwijany przez Microsoft. Edytor ten jest z założenia edytorem minimalistycznym, zawierającym w swej podstawowej formie tylko niezbędne narzędzia, jednakże przez bycie projektem o otwartym źródle, jest on łatwo rozszerzalny dzięki szerokiej palecie wtyczek tworzonych przez podmioty komercyjne oraz członków społeczności. Szeroka gama rozszerzeń pozwala użytkownikom Visual Studio Code na dużą swobodę dostosowywania tego edytora pod swoje własne preferencje ergonomii pracy, jak i różnorodne projekty w większości języków programowania, przekształcając go w pełnoprawne interaktywne środowisko programistyczne, posiadające możliwość pracy ze zdalnymi zasobami.

Visual Studio Code posiada wbudowaną obsługę funkcji edycji kodu dla najpopularniejszych języków programowania takich jak:

- Autouzupełnianie kodu
- Informacje dotyczące parametrów
- Refactoring
- Analiza kodu

które to pozwalają na szybszą i bardziej efektywną pracę.

### **4.3. Google Colaboratory**

Darmowa usługa chmurowa środowiska notatników Jupyter, pozwalająca na pisanie, wykonywanie i współdzielenie skryptów języka Python z poziomu przeglądarki internetowej. Główną zaletą tej usługi jest możliwość użytkowania zasobów zewnętrznego serwera, posiadającego znaczące moce obliczeniowe oraz wyspecjalizowany hardware do uczenia maszynowego, dając dostęp dla szerszej grupy ludzi do nauki i rozwoju w dziedzinie uczenia maszynowego i analizy danych, którzy nie mają dostępu do odpowiednich mocy obliczeniowych, wymaganych przez bardziej skomplikowane modele sztucznej inteligencji.

## **ROZDZIAŁ 5**

# **Implementacja programu**

W rozdziale piątym zostały przedstawione najważniejsze fragmenty sposobu implementacji programu.

### **5.1. Podrozdział**

...

## **Zakończenie**

Tu podsumowanie z realizacji założeń, możliwości dalszej pracy

## **Bibliografia**

## **Spis tabel**

## **Spis rysunków**

## **Spis listingów**