

Spis treści

Wstęp	2
Cel i zakres pracy	3
1 Rozdział z dziedziny pracy - część teoretyczna	4
1.1 Rys historyczny	4
1.2 Przegląd istniejących rozwiązań	4
2 Zakres użytych technologii i opis wykorzystywanych narzędzi	5
2.1 OpenCV	5
2.2 Visual Studio Code	5
2.3 Git	5
2.4 GitHub	5
3 Realizacja projektu	6
3.1 Identyfikacja aktorów	6
3.2 Wymagania funkcjonalne i нефункционалне	6
3.2.1 Wymagania funkcjonalne	6
3.2.2 Wymagania нефункционалне	6
4 Podsumowanie	7
Bibliografia	8
Spis rysunków	9
Spis tabel	10
A. Załączniki	11

Wstęp

Przetwarzanie obrazów to dziedzina informatyki zajmująca się automatyzacją zadań związanych z obrazami. Obejmuje ona szeroki zakres zagadnień, takich jak sztuczna inteligencja, uczenie maszynowe, analiza danych i grafika komputerowa.

Oprogramowanie do przetwarzania obrazów jest wykorzystywane w wielu różnych dziedzinach, takich jak medycyna, przemysł, bezpieczeństwo i rozrywka. W medycynie jest wykorzystywane do diagnostyki, obrazowania medycznego i chirurgii. W przemyśle jest wykorzystywane do kontroli jakości, automatyzacji i inspekcji. W bezpieczeństwie jest wykorzystywane do monitorowania, rozpoznawania twarzy i identyfikacji. W rozrywce jest wykorzystywane do tworzenia efektów specjalnych, animacji i gier wideo.

Jedną z popularnych bibliotek do przetwarzania obrazów jest OpenCV. OpenCV to darmowy, otwartoźródłowy pakiet oprogramowania dla systemów operacyjnych Windows, macOS i Linux. Zapewnia szeroki zakres funkcji do przetwarzania obrazów, w tym: podstawowe operacje takie jak filtrowanie, skalowanie i konwersja formatów, wykrywania i rozpoznawania obiektów czy obsługę algorytmów do analizy wideo.

W ramach czasu, który został poświęcony na niniejszy projekt zakres możliwości ograniczono do prostych operacji przetwarzania obrazów bez uwzględniania wideo oraz wszelkiego rodzaju danych oprócz samych obrazów.

OpenCV jest często wykorzystywany do tworzenia oprogramowania do przetwarzania obrazów. Jednak jego interfejs użytkownika jest dostosowany do programistów, a nie do użytkowników końcowych. Oznacza to, że użytkownicy końcowi muszą mieć podstawowe umiejętności programowania, aby korzystać z oprogramowania stworzonego w oparciu o OpenCV.

W tej pracy naukowej proponujemy nowe oprogramowanie do przetwarzania obrazów oparte na bibliotece OpenCV. Oprogramowanie to posiada funkcjonalny interfejs graficzny, który ułatwia użytkownikom końcowym korzystanie z niego.

Cel i zakres pracy

Celem tej pracy jest zaprojektowanie i implementacja oprogramowania do przetwarzania obrazów opartego na bibliotece OpenCV z funkcjonalnym interfejsem graficznym.

Zakres pracy objął: analizę istniejących rozwiązań na podstawie której stwierdzono, że nie ma aktualnego i gotowego produktu który realizuje dokładnie założenie zastąpienia programistycznego podejścia do obsługi biblioteki OpenCV. Jest wiele rozwiązań które realizuje część biblioteki lub jej obsługa nie pozwala na przetwarzanie obrazów w niedestrukcyjny sposób bez dodatkowego wysiłku od użytkownika jak używanie warstw czy inkrementalnych zapisów.

Jako platformę wybrano komputer osobisty z powodu znaczącej mocy obliczeniowej, dostępu do dużej ilości pamięci RAM oraz dostępu do operacji procesorów x86. [1] Jako docelowy system operacyjny wybrano Microsoft Windows 11 ponieważ jest to system używany przez autora tej pracy. Do tworzenia aplikacji skorzystano z języka C# oraz Windows Presentation Framework.

Pomysł na ten projekt powstał dzięki doświadczeniu autora pracy w używaniu różnych systemów opartych o edytory typu węzłowego jak Adobe Substance Designer, edytor Blueprintów w Unreal Engine oraz edytor shaderów czy geometry nodes w Blenderze. Ten typ interfejsu pozwala na interakcję użytkownika z wizualnym przedstawieniem jakichś metod, funkcji czy bloków logicznych pozwalających na przesyłanie wartości w podobny sposób jak można robić to w kodzie, ale zdecydowanie bardziej przystępne dla osób które nie mają odpowiedniego doświadczenia w programowaniu.

1 Rozdział z dziedziny pracy - część teoretyczna

1.1 Rys historyczny

Z plików dla Więclawa

1.2 Przegląd istniejących rozwiązań

2 Zakres użytych technologii i opis wykorzystywanych narzędzi

Zastosowane technologie:

- **Otwarty kod źródłowy:** OpenSource jest spoko

2.1 OpenCV

2.2 Visual Studio Code

2.3 Git

2.4 GitHub

3 Realizacja projektu

3.1 Identyfikacja aktorów

Lista aktorów?

- **Użytkownik:** Program powinien być przystępny dla każdego

3.2 Wymagania funkcjonalne i нефunkcjonalne

Oprogramowanie opisane w tej pracy to **NoodleCV**. Następne podrozdziały przedstawia wymagania z jakimi trzeba było się zmierzyć w trakcie tworzenia aplikacji.

3.2.1 Wymagania funkcjonalne

Wymagania funkcjonalne?

- Przetwarzanie obrazów

3.2.2 Wymagania нефunkcjonalne

Неfunkcjonalne wymagania?

- Łatwość użytkowania

4 Podsumowanie

Na koniec należy podsumować, co było celem pracy. Czy cel ten został osiągnięty. Jakie są możliwe wnioski. Czy są dalsze możliwości rozwoju.

Bibliografia

- [1] OpenCV. *OpenCV CPU optimizations*. URL: <https://github.com/opencv/opencv/wiki/CPU-optimizations-build-options>. (dostęp: 07.01.2024).

Spis rysunków

Spis tabel

A. Załączniki